

ADAPTIVE WEB SITES

VISIT...

LANZAROTE
Caliente.COM

Frontiers in Artificial Intelligence and Applications

Volume 170

Published in the subseries

Knowledge-Based Intelligent Engineering Systems

Editors: L.C. Jain and R.J. Howlett

Recently published in KBIES:

- Vol. 149. X.F. Zha and R.J. Howlett (Eds.), Integrated Intelligent Systems for Engineering Design
- Vol. 132. K. Nakamatsu and J.M. Abe (Eds.), Advances in Logic Based Intelligent Systems – Selected Papers of LAPTEC 2005

Recently published in FALA:

- Vol. 169. C. Branki, B. Cross, G. Díaz, P. Langendörfer, F. Laux, G. Ortiz, M. Randles, A. Taleb-Bendiab, F. Teuteberg, R. Unland and G. Wanner (Eds.), Techniques and Applications for Mobile Commerce – Proceedings of TAMoCo 2008
- Vol. 168. C. Riggelsen, Approximation Methods for Efficient Learning of Bayesian Networks
- Vol. 167. P. Buitelaar and P. Cimiano (Eds.), Ontology Learning and Population: Bridging the Gap between Text and Knowledge
- Vol. 166. H. Jaakkola, Y. Kiyoki and T. Tokuda (Eds.), Information Modelling and Knowledge Bases XIX
- Vol. 165. A.R. Lodder and L. Mommers (Eds.), Legal Knowledge and Information Systems – JURIX 2007: The Twentieth Annual Conference
- Vol. 164. J.C. Augusto and D. Shapiro (Eds.), Advances in Ambient Intelligence
- Vol. 163. C. Angulo and L. Godo (Eds.), Artificial Intelligence Research and Development
- Vol. 162. T. Hirashima et al. (Eds.), Supporting Learning Flow Through Integrative Technologies
- Vol. 161. H. Fujita and D. Pisanelli (Eds.), New Trends in Software Methodologies, Tools and Techniques – Proceedings of the sixth SoMeT_07
- Vol. 160. I. Maglogiannis et al. (Eds.), Emerging Artificial Intelligence Applications in Computer Engineering – Real World AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies
- Vol. 159. E. Tyugu, Algorithms and Architectures of Artificial Intelligence
- Vol. 158. R. Luckin et al. (Eds.), Artificial Intelligence in Education – Building Technology Rich Learning Contexts That Work

ISSN 0922-6389

Adaptive Web Sites

A Knowledge Extraction from Web Data Approach

Juan D. Velásquez

*Department of Industrial Engineering, School of Engineering and Science,
University of Chile, Santiago, Chile*

and

Vasile Palade

Oxford University Computing Laboratory, Oxford, United Kingdom

IOS
Press

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

© 2008 The authors and IOS Press.

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without prior written permission from the publisher.

ISBN 978-1-58603-831-1

Library of Congress Control Number: 2007943828

Publisher

IOS Press

Nieuwe Hemweg 6B

1013 BG Amsterdam

Netherlands

fax: +31 20 687 0019

e-mail: order@iospress.nl

Distributor in the UK and Ireland

Gazelle Books Services Ltd.

White Cross Mills

Hightown

Lancaster LA1 4XS

United Kingdom

fax: +44 1524 63232

e-mail: sales@gazellebooks.co.uk

Distributor in the USA and Canada

IOS Press, Inc.

4502 Rachael Manor Drive

Fairfax, VA 22032

USA

fax: +1 703 323 3668

e-mail: iosbooks@iospress.com

LEGAL NOTICE

The publisher is not responsible for the use which might be made of the following information.

PRINTED IN THE NETHERLANDS

To our lovely wives Queny and Dana

This page intentionally left blank

Contents

1	Introduction	1
1.1	The World Wide Web	2
1.2	Towards new portal generation	5
1.3	Structure of the book	7
2	Web data	9
2.1	Web's Operation	10
2.2	The information behind the clicks	13
2.2.1	Session reconstruction process	15
2.2.2	Finding real sessions	18
2.3	The information contained in a web page	19
2.3.1	Web page content	19
2.3.2	Web page links	21
2.4	Summary	23
3	Knowledge discovery from web data	25
3.1	Overview	26
3.2	Data sources and cleaning	28
3.3	Data consolidation and information repositories	30
3.4	Data Mining	32
3.4.1	Motivation	32
3.4.2	Data Mining techniques	33

CONTENTS

3.4.2.1	Association rules	33
3.4.2.2	Classification	34
3.4.2.3	Clustering	34
3.5	Tools for mining data	37
3.5.1	Artificial Neural Networks (ANN)	37
3.5.2	Self-Organizing Feature Maps (SOFMs)	41
3.5.3	K-means	43
3.5.4	Decisions trees	44
3.5.5	Bayesian networks	46
3.5.6	K-Nearest Neighbor (KNN)	48
3.5.7	Support vector machines (SVMs)	50
3.6	Using data mining to extract knowledge	53
3.7	Validation of the extracted knowledge	55
3.8	Mining the web	55
3.9	Summary	56
4	Web information repository	59
4.1	A short history of data storage	60
4.2	Storing historical data	62
4.3	Information systems	63
4.4	Data Mart and Data Warehouse	65
4.4.1	The multidimensional analysis	67
4.4.2	The Cube Model	70
4.4.3	The Star Model	72
4.4.4	The Extraction, Transformation and Loading Process	75
4.4.4.1	Extraction	75
4.4.4.2	Transformation	76
4.4.4.3	Loading	77
4.5	Web warehousing	77

CONTENTS

4.6	Information repository for web data	79
4.6.1	Thinking the web data in several dimensions	80
4.6.2	Hyper cube model for storing web data	82
4.6.3	Star model for storing web data	84
4.6.4	Selecting a model for maintaining web data	85
4.6.5	ETL process applied to web data	86
4.6.5.1	Processing web page text content	87
4.6.5.2	Processing the inner web site hyperlinks structure . . .	87
4.6.5.3	Processing the web logs	88
4.7	Summary	90
5	Mining the Web	93
5.1	Mining the structure	94
5.1.1	The HITS algorithm	95
5.1.2	The Page Rank algorithm	98
5.1.3	Identifying web communities	101
5.2	Mining the content	102
5.2.1	Classification of web page text content	103
5.2.2	Clustering for groups having similar web page text content . . .	105
5.2.3	Some applications	106
5.2.3.1	WEBSOM	106
5.2.3.2	Automatic web page text summarization	107
5.2.3.3	Extraction of key-text components from web pages . .	108
5.3	Mining the usage data	109
5.3.1	Statistical methods	110
5.3.2	Clustering the user sessions	110
5.3.3	Classification of the user behavior in a web site	112
5.3.4	Using association rules for discovering navigation patterns . . .	113

CONTENTS

5.3.5	Using sequence patterns for discovering common access paths	114
5.3.6	Some particular implementations	115
5.3.6.1	Web query mining	115
5.3.6.2	Prefetching and caching	116
5.3.6.3	Helping the user's navigation in a web site	118
5.3.6.4	Improving the web site structure and content	119
5.3.6.5	Web-based adaptive systems	120
5.4	Summary	121
6	Web-based personalization systems	125
6.1	Recommendation Systems	126
6.1.1	Short historical review	127
6.1.2	Web-based recommender systems	129
6.1.2.1	Web recommender systems, particular approaches and examples	132
6.2	Systems for personalization	133
6.2.1	Computerized personalization	134
6.2.2	Effectiveness of computerized personalization systems	136
6.2.3	Computerized personalization approaches	137
6.3	Web personalization	139
6.3.1	Aspects of web personalization privacy	141
6.3.2	Main approaches for web personalization	143
6.3.3	Privacy aspects of web personalization privacy	144
6.4	Adaptive web-based systems	147
6.4.1	A short introduction	148
6.4.2	Elements to take into account	150
6.4.3	Web site changes and recommendations	151
6.4.4	Adaptive systems for web sites	153
6.5	Summary	154

CONTENTS

7	Extracting patterns from user behavior in a web site	157
7.1	Modelling the web user behavior	158
7.2	Web data preparation process	162
7.2.1	Comparing web page contents	163
7.2.2	Comparing the user navigation sequences	166
7.3	Extracting user browsing preferences	169
7.3.1	Comparing user browsing behavior	169
7.3.2	Applying a clustering algorithm for extracting navigation patterns	170
7.4	Extracting user web page content preferences	173
7.4.1	Comparing user text preferences	174
7.4.2	Identifying web site keywords	175
7.5	Summary	177
8	Acquiring and maintaining knowledge extracted from web data	179
8.1	Knowledge Representation	180
8.1.1	Fundamental roles of knowledge representation	180
8.1.2	Rules	182
8.1.3	Knowledge repository	183
8.2	Representing and maintaining knowledge	183
8.3	Knowledge web users	185
8.4	A framework to maintain knowledge extracted from web data	186
8.4.1	Overview	186
8.4.2	The Web Information Repository	188
8.4.3	The Knowledge Base	189
8.4.3.1	Pattern Repository	190
8.4.3.2	Rule Repository	191
8.5	Integration with adaptive web sites	193
8.6	Summary	194

CONTENTS

9 A framework for developing adaptive web sites	195
9.1 The adaptive web site proposal	196
9.2 Selecting web data	197
9.3 Extracting information from web data	200
9.3.1 The star model used for the creation of the WIR	201
9.3.2 Session reconstruction process	201
9.3.3 Web page content preprocessing	206
9.4 Applying web mining techniques	208
9.4.1 Analyzing the user browsing behavior	208
9.4.1.1 Applying statistics	208
9.4.1.2 Using SOFM for extracting navigation patterns	209
9.4.1.3 Using K-means for extracting navigation patterns	212
9.4.2 Analyzing user text preferences	213
9.5 Using the extracted knowledge for creating recommendations	217
9.5.1 Offline recommendations	217
9.5.1.1 Structure recommendations	217
9.5.1.2 Content recommendations	219
9.5.2 Online recommendations	220
9.5.3 Testing the recommendation effectiveness	220
9.5.3.1 Testing offline structure recommendation	221
9.5.3.2 Testing offline content recommendation	225
9.5.3.3 Testing online navigation recommendation	226
9.5.4 Storing the extracted knowledge	229
9.5.4.1 Pattern Repository	230
9.5.4.2 Rules for navigation recommendations	231
9.6 Summary	233
In place of conclusions	237

CONTENTS

Bibliography	241
---------------------	------------

This page intentionally left blank

List of Figures

2.1	The web server - web browser interaction	10
2.2	A typical web log file	14
2.3	Sessionization process suing a time based heuristic	17
2.4	A web page represented in the vector space model	21
2.5	A web community as a directed graph	22
3.1	The stages of the KDD process.	27
3.2	A biological neuron representation	37
3.3	A multi-layer Artificial Neural Network	40
3.4	SOFM topologies	43
3.5	K-means after the first iteration	44
3.6	A decision tree for scarf purchases	46
3.7	A Bayesian network for clothes purchases	48
3.8	KNN classification example	50
3.9	The separating surface returned by an SVM	52
4.1	A simple sale business report	69
4.2	A cube model for a sale data mart	71
4.3	A star model for a sale data mart	73
4.4	A snowflake model for a sale data mart	74
4.5	A generic data web warehouse architecture	80
4.6	A generic cube model for web data	82

FIGURES

4.7	A generic star model for web data	84
4.8	Data staging area for processing web logs	89
5.1	A simple web-graph for a web community	97
5.2	A web community identification by using Maximum Flow method . . .	102
6.1	Web personalization approaches	138
6.2	User model adaptation	149
7.1	User behavior vector's creation	161
7.2	Including the importance of special words in the vector space model . .	166
7.3	A web site with two navigation sequences	167
7.4	Neighborhood of neurons in a thoroidal Kohonen network	171
8.1	Knowledge representation using rules	182
8.2	A framework to acquire and maintain knowledge extracted from web data	187
8.3	A generic Knowledge Base Structure	189
8.4	A generic Pattern Repository model	190
8.5	Sample of pseudo-code from Rule Repository	192
9.1	A framework for constructing adaptive web sites	197
9.2	Home page of the virtual bank	199
9.3	Bank's web site layout	200
9.4	Data Mart for web information using the star model	202
9.5	A raw web log file from the bank web site	202
9.6	Data staging area for session reconstruction process	203
9.7	A regular expression for processing web logs	204
9.8	SQL pseudocode for sessionization	206
9.9	Similarity measure among web pages	207
9.10	Clusters among user behavior vectors	211

FIGURES

9.11 Clusters of important page vectors	214
9.12 Clusters of user behavior vectors using 70% of the data	228
9.13 Percentage of acceptance of online navigation recommendations	229

This page intentionally left blank

List of Tables

3.1	Scarf's purchase behavior used to create a decision tree	45
9.1	Summary of the statistics of the bank web site	209
9.2	The top ten pages ranked by average time spent per page	209
9.3	The ten most visited pages	210
9.4	Bank web site pages and their content	210
9.5	User behavior clusters	212
9.6	K-means user behavior clusters	213
9.7	Important page vectors clusters	215
9.8	The 8 most important words per cluster	215
9.9	A part of the discovered keywords	216
9.10	New bank web site pages and their content	221
9.11	Navigation behavior searching the real web page A	223
9.12	Navigation behavior searching the real web page B	223
9.13	Navigation behavior searching a false web page	224
9.14	Usability test for the web site hyperlink structure	224
9.15	Testing the web site keyword effectiveness	226
9.16	User behavior clusters using 70% of the data	227
9.17	An example of rule operation	232

This page intentionally left blank

Acknowledgments

“Knowledge is in the end based on acknowledgement”

Ludwig Wittgenstein

We are grateful to many friends and colleagues for giving us their particular points of view, commentaries, advice and encouragement, making it possible for us to fulfill our dream of finishing this book. Unfortunately the list of contributors is so long that it would be difficult to include all of them. However, we would like to recognize some of those whose assistance and help were decisive during this project.

First of all, we want to thank Professor Andrés Weintraub and Dr. Rafael Epstein for giving us the necessary support to finish the book. Also we are grateful to Professor Ricardo Baeza-Yates for his valuable commentaries about the book’s content and Dr. Michael A. Berry, who provided his critical expertise to the book from an industry point of view. Also we would like to thank Dr. Luis Vargas for his enthusiasm and encouraging us to write the book and Mr. Anthony Tillet for his editorial assistance.

Finally, we are very grateful to the Millennium Institute Systems Complex Engineering, which partially funded this work.

Juan D. Velásquez

Santiago, Chile.

Vasile Palade

Oxford, United Kingdom.

December 2007

This page intentionally left blank

Foreword

This book can be presented in two different ways. Based on the title, we can think that this book introduces a particular methodology to build adaptive Web sites. To achieve that, the book first introduces all the concepts needed to understand and apply the proposed methodology.

The alternate view is as a book that presents the main concepts behind Web mining and then applies them to adaptive Web sites. In this case, adaptive Web sites is the case study to exemplify the tools introduced in the text. In the following paragraphs I use the later view to give the reader a quick tour about the book's content.

The authors start by introducing the Web and motivating the need for adaptive Web sites, which is mainly commercial. The second chapter introduces the main concepts behind a Web site: its operation, its associated data and structure, user sessions, etc. Chapter three explains the Web mining process and the tools to analyze Web data, mainly focused in machine learning. The fourth chapter looks at how to store and manage data, as in a popular Web site we can generate gigabytes of logs in a short period of time. Chapter five looks at the three main and different mining tasks: content, links and usage. The following chapter covers Web personalization, a crucial topic if we want to adapt our site to specific groups of people. Chapter seven shows how to use information extraction techniques to find user behavior patterns. The subsequent chapter explains how to acquire and maintain knowledge extracted from the previous phase. Finally, chapter nine contains the case study where all the

previous concepts are applied to present a framework to build adaptive Web sites.

In other words, the authors have taken care of writing a self-contained book for people who want to learn and apply personalization and adaptation in Web sites. This is commendable considering the large and increasing bibliography in these and related topics. The writing is easy to follow and although the coverage is not exhaustive, the main concepts and topics are all covered. The level is fair and appropriate for a last year undergraduate or a graduate course.

In a personal digression, I am really glad to see a book where one of the authors was one of my students and, hence, feel that I have contributed with a small grain of sand to this book. In fact, having wrote a few books, I know how hard it is to write one and how much motivation you need to finally see the printed outcome. Finally, without doubt, nowadays e-commerce is one of the main sales channels and clearly learning from your Web customers has a significant commercial value. Sadly, not too many Web sites use the full potential of Web mining and I hope this book makes an important contribution to change that.

Ricardo Baeza-Yates
VP for Europe and Latin America
Yahoo! Research
Barcelona, Spain & Santiago, Chile
December 2007

Chapter 1

Introduction

*The beginning of knowledge is the discovery
of something we do not understand.*

Frank Herbert

The World Wide Web or **the Web** [24] has revolutionized communications for governments, individuals and companies. Its impact in our society has been so significant that some authors compare it to the invention of the wheel or the discovery of fire.

The Web is now a massive channel for worldwide diffusion and information exchange; companies have responded to this challenge in several ways of which the most common has been the creation of corporative web sites that show specific information about the company, as a kind of “*virtual business card*”. The sites were soon integrated into the internal company system, for example, sales. From this point on, the design and construction of a web site became a complex task with new business models appearing in the digital market. For many companies/institutions, it is no longer sufficient to have a web site and provide high quality products or services. The difference between the success and failure of an e-business may be given to the potential of the web site to attract and retain visitors. This potential is determined by the web site’s content, design, and not least technical aspects, for example, the

time to load pages when compared to other sites.

In this new environment, the fight to retain customers or catch new ones is a decisive factor for a company's success in the digital market. While web site visits are a key basic datum, there is a crucial distinction to be drawn between a visitor and a customer. A *visitor* is a person that enters the web site without leaving information about him or her, except pages visited and the time spent on them. A *customer* is a person identified by the company, for whom some personal data are known, such as age, sex, birthday, etc., and provides identification, like a user/password, to enter the *non public* part of web site. Finally, *users* include both visitors and customers as well as other specific groups such as web masters etc.

To remain competitive, a company needs an up-to-date web site that offers the information the users are looking for in an easily accessible way. At best, the web site should perform efficiently, automatically and online. However, the reality is often quite different. In too many cases, the web site's structure does not help the user find the desired information, even if it is somewhere in the site.

1.1 The World Wide Web

The origin of the Internet goes back to the period of the cold war when the United States (USA) and the Soviet Union (USSR) were competing for technological superiority. As information is vital during armed conflict, the USA government devised an interconnected computer system which put a premium on speed and safety for information distribution. Further more the network was designed to be able to sustain the loss of a group of computers and continue to perform its functions.

The Pentagon, through its Defense Advanced Research Projects Agency (DARPA) financed the beginning of experimental trials. In 1969, the first node of the network ARPANET was opened, at the University of California in Los Angeles. The network began to expand, and a second node was installed at the Stanford

Research Institute (SRI).

The original but somewhat (from today's perspective) primitive idea of connecting computers with other computers was transformed into a new and powerful approach, computer networks linking computers with others. This interconnection gave the net its name - **Inter-Net**. From its beginnings, the Internet did something more than simply allow computers to communicate - it gave way to interactions between persons through new tools such as the electronic mail (e-mail). During the Internet's initial years use was restricted to a few privileged people, who were able to understand the complex instructions required to use Internet services.

In the early 1990s, Tim Berners-Lee, a researcher at the "Conseil Européen pour la Recherche Nucleaire" (CERN) in Switzerland, developed the idea of shared hypertext [24]. It was the birth of the Web. From this moment, the form in which we, as human beings, communicate with one another significantly changed.

"The World Wide Web (W3) is the universe of network-accessible information, an embodiment of human knowledge. It is an initiative started at CERN, now with many participants. It has a body of software, and a set of protocols and conventions. W3 uses hypertext and multimedia techniques to make the web easy for anyone to roam, browse, and contribute to". -Tim Berners-Lee (1993)

The Web is changing everything; from how we do businesses, to the way we relate to one another and not least, its effect on the economy in general and how knowledge is acquired and generated in particular. An obvious example is how pupils and students do their homework. Primary school students use computers connected to the Internet, working with complex systems like search engines and sharing their experiences by e-mail and forums. To put simply as "the is the network, the network is the computer"¹ [19].

A major Web linked transformation concerning the relationship between

¹SUN's slogan

providers and consumers of goods and services. Traditionally, between both actors, there is a chain of intermediaries involved in the distribution of goods and services. Every link in the chain adds to the product price but not necessarily to its value. As consumers are unlikely to buy in bulk or pay the premiums for direct delivery, the chain and its steps are necessary for distribution. However it's unlikely that consumers know about all individual product suppliers so their purchasing decision may not be well informed. The Web solves the first situation - consumers can directly request a given product or product concept, perhaps motivating the interest of the manufacturer to produce it. As important, the Web provides substantial amounts of information - perhaps more than he or she needs - helping in part the challenges of information asymmetry.

These changes are behind the birth of new web-based businesses. At the end of the twentieth century, many web-based companies were overvalued and were without real capital sustenance. This occurred for example on the NASDAQ² [122] when the share value of some of these companies fell to a third of their highest value achieved in the New York stock-market.

Despite all the problems and disappointments that occurred during the first stages of the Web, it is now agreed that companies and markets must become more virtual. Many companies have created complex sale systems that use web technology to offer product alternatives to potential customers.

In this new environment influenced by the Web, the fight to retain old customers and to attract new ones is decisive for the survival of a company in the digital market. Many firms have failed in their attempt to remain Web competitive. The reasons for failure are plentiful, but the general consensus is that too many firms are incapable of adapting quickly to the changes that the Web imposes. This is a clear symptom of declining competitiveness, leading to a loss of market share and possible financial bankruptcy.

²National Association of Securities Dealers Automated Quotation System

Market share depends on new customers and retaining the existing ones [245]. For this, it is vital to understand a customer's purchasing behavior in web terms. This is a complex problem and has been approached in many ways. Experts in electronic commerce have noticed that the key issue is the content of the web site [172], which often shows information that might be irrelevant for the visitor, therefore hindering rapid information search. Additionally, as is well known, a user, on average, visits three web pages per site before deciding whether the web site is useful to him or not [151].

The different types of electronic web businesses, which include Business to Business (B2B), Business to Consumer (B2C), Peer to Peer (P2P) and any variation of these models, require a more sophisticated portal which can auto configure its structure and content to satisfy the requirements of potential users [181, 180, 228].

1.2 Towards new portal generation

What is the ideal structure and content of a web site? While there is no simple answer to this question - the minimum requirement for a web site is that it facilitates the users search for information. In many cases, the web site structure does not reveal the desired information easily, even though the information is to be found somewhere in the web site.

A good web design must take account of the following; that,

- Different users have distinct goals.
- The behavior of users changes over time.
- Sites must be restructured as they grow to meet current needs, typically by accumulating pages and links.

A key feature of successful web sites is the ability to provide the right content

at the right moment for the user. Development should take past history into account and improvements made to the web site's structure and content, by preparing hints to help users to find what they are looking for quickly and efficiently.

The **Adaptive Web Site** (AWS) is the principal concept behind new portal generation [179, 238]. Based on user behaviour, AWSs can implement changes to the current web site structure and content. Changes that alter the web site, run the risk of user rejection, and so it is advisable to implement them as recommendations for the visitor or the expert who maintains the web site (web masters).

Web site recommendations can be grouped into two categories: online and offline. The former is provided in real time by an automatic system; for example, navigation recommendations about the next page to visit, based on the previously visited pages. These are performed manually and correspond to physical changes in the web site structure and/or content. Both kinds of recommendations should understand the user's browsing behavior and his/her content preferences.

Web Intelligence (WI) [155] designates a set of research topics in Artificial Intelligence (AI) (e.g., knowledge representation, planning, knowledge discovery and data mining, intelligent agents, social network intelligence) that have and will be employed in current and future generations of web-empowered products, systems, services, and activities. It represents one of the most important and promising Information Technology (IT) research fields in the Web era.

A specific WI area is the study of the user behavior in a web site [234], which has the objective of better understanding user preferences and from this to develop a more attractive web site structure and content. This is not a trivial task and there are a significant number of research laboratories worldwide working in WI and related areas. A major research area is web usage mining [136], which contributes to the analysis and exploration of web sites using information that users leave behind when navigating a particular web site. Many algorithms and systems have already been proposed for this purpose [83].

1.3 Structure of the book

The main purpose of this book is to provide both an introduction and in-depth illustrations about the different stages of AWS construction. The first five chapters present the theory behind the construction of web-based systems which personalize the user's web site experience. The remaining chapters will show the practical application of the theory and algorithms for the development of an AWS in a real world case.

Chapter 2 examines the web site's operation, the nature of data which originates in the Web (web data) and the techniques used for cleaning and preprocessing web data for pattern extraction tasks.

The Knowledge Discovery from Databases (KDD) process is introduced in chapter 3 which also shows how KDD methods can be used to extract knowledge from web data. The chapter concludes with a survey of the main data mining tools used to process web data.

Each change applied in a web site can provide important insights into user information preferences. However, as repositories which contain web site changes and related data is rarely maintained, the success/failure history is lost. In chapter 4, a Web Information Repository (WIR) is proposed using data warehouse architecture for storing web data and information extracted during the web site operation.

Chapter 5 presents a short survey of the main issues and associated algorithms for mining web data. The chapter concludes by showing some practical applications of web mining algorithms in real world cases.

The following chapter, 6, is a practical review of the main approaches in web-based systems for the personalization of a web site for users. This chapter also gives details about the structure and the operation of AWSs.

An important step in AWS construction is the development of web mining algo-

rithms and models for analyzing the user's behavior on the web site. Chapter 7 tackles this issue and presents a practical approximation for the pattern and knowledge extraction process from web data. The newly extracted knowledge must be maintained to allow consultation by human users or artificial systems. Chapter 8 presents an original framework for representing, acquiring and storing the knowledge.

The final chapter - chapter 9 - shows a practical real-world application of the theories, methods and algorithms presented and developed throughout the book. Here, real data is collected from a web site that belongs to a virtual bank, i.e., a bank that has no physical branches and where all transactions are undertaken electronically.

Chapter 2

Web data

*Data is not information, Information is not knowledge,
Knowledge is not understanding, Understanding is not wisdom.*

Cliff Stoll & Gary Schubert

This chapter concentrates on data which originated on the Web, known as web data, which may have different sources, formats thus usually accompanied by high levels of extraneous components. This data requires preprocessing and cleaning operations to prepare for pattern discovery tasks.

Among different potential web data sources, special attention must be paid to web logs, web pages and web site hyperlinks structure as they are used as inputs for most web mining algorithms.

This chapter also examines the nature of the web data to show the typical problems encountered in the transformation of data into feature vectors, to be used for pattern extraction.

2.1 Web's Operation

We provide a basic survey of web operations . At its simplest the web operations are based on the client/server paradigm [95], by which a web client (browser) requests services from a web server. This interaction begins when a user accesses a web site by writing the URL¹ in the web browser or by clicking a hyperlink in another web page . Immediately, a request is sent to the web server using the Hyper Text Transfer Protocol (HTTP)² and the web server returns the requested object.

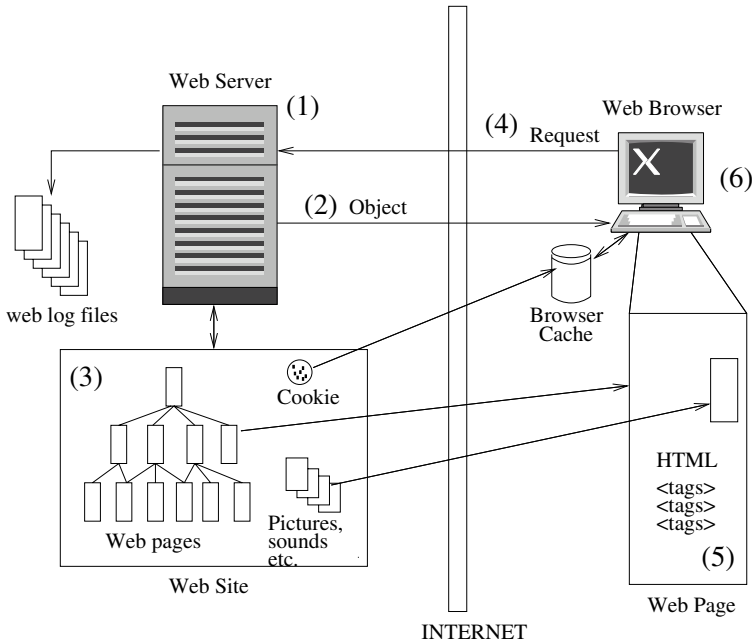


Figure 2.1: The web server - web browser interaction

Fig. 2.1 shows the interaction between the web server and the browser. The web server (1) is a program that runs in the background that receives the request (4)

¹Uniform Resource Locator

²<http://www.w3.org/Protocols/>

through a specific port, usually the port 80. This program manages a structured set of files, known as a web site (3). Part of these files are web pages that contain links to other types of data files, such as pictures, sounds, movies or other pages. The main function of the web server is to fulfill web pages requests. A web page (5) is written using the Hyper Text Markup Language (HTML) ³. This consists of a sequence of orders called tags concerning methods on how to display in the user screen the objects (2) requested by the browser (6) and how to retrieve other objects from web server . These orders are interpreted by the browser which show the objects on the user screen.

Once the document has been read by the browser, the specific inside tags can be interpreted. When the browser which interprets the tags find a reference about an object, for example an image, the HTTP transfers it to the browser. The process is completed when the last tag is interpreted and the page is shown to the user.

The browser speed of unfolding the page depends on several factors. The most important is the bandwidth of the existing connection between browser and server, which depends on the ISP⁴ and the device used for the connection, e.g., a conventional modem or an Asymmetric Digital Subscriber Line (ADSL) system.

The second factor is the speed of the computer, particularly with respect to the browser. The contents of many pages are rather complex, for instance they may refer or contain Java Applets, Java Scripts, Dynamic HTML, etc. In these cases, the computer's resources, like CPU, RAM or Hard Disk, may not be enough. For many commercial sites, this can be a great problem. In fact, some authors [40, 172] recommend that pages be sent to browsers should only contain simple elements and standards, transferring the more complex processing to the server.

However web servers are unlikely to have sufficient resources themselves to deal with the requests submitted by a large number of users and must rely on advanced

³<http://www.w3.org/MarkUp/>

⁴Internet Service Provider

architectures such as the multiple tier model [95]. This model separates the three logical layers - namely Interface, Logic and Data -physically in the Client/Server model. The logic layer where the complex algorithms that make up the business core are located, is usually housed in a powerful stand alone computer, where the web server and web site are usually installed.

An alternative is to increase the transfer rate and use *cache* for storing the web objects retrieved. The cache is a special memory buffer, sometimes in the computer's main memory or the client's directory, in which recently collected objects from the web server are transiently stored. If the same page is visited again, the objects are not requested from the web server, because these are in the cache.

A variant of this client side cache method is by using a corporate cache that contains the web pages visited by all the institution's users. When a browser requires a specific web page, first it consults the corporative cache and, if the page is already stored, does not transfer it from the real web site. Otherwise, the page is transferred and stored in the cache. The corporate cache is usually implemented by a proxy server in a special network computer.

However neither cache scheme show the web page request and with some proxy configurations hides the entire network behind the proxy IP address. With these difficulties more invasive detection methods need have to be applied, such as the agents or cookies. Agents are programs running on the browser whose function is to send the complete user navigation sequence to the server. A cookie is an identification mechanism which consists of a small code sent from the server to the browser which normally contains a short description about the session with an identification number. The browser saves it in its cache and when the web site is revisited it sends the cookie to the server and the user is identified.

2.2 The information behind the clicks

In order to anticipate an user's decisions at a web site, it is necessary to make an in-depth analysis of his or her behavior [233], which would include, for example, the pages visited, the time spent on each page, the products chosen, etc.

The web log registers contain information about the user's browsing behavior, in particular the page navigation sequence and the time spent on each page visited. The data from web logs is based on the structure given by W3C⁵. activity. These logs can contain millions of registers although most are unlikely to hold relevant information.

When a web page is accessed, the HTML code (with the web page tags that refer to various web objects), is interpreted by the browser. A register is then created in the web log file for the accessed page as well as for each object referred on that page.

It is important to note that even if the user makes only one click, the transmitted objects from server to browser could be more than one. A short description of the objects is loaded on to the web logs, showing many of the registers with irrelevant data inside [229].

Fig. 2.2 shows a typical web log file. The file structure is a configurable parameter at the web site with normally similar content. It is defined by:

IP address; the internet host address. By using the *reverse address lookup protocol*, it is possible to find the client's domain name.

Identity; identity information supplied by the customer.

Authuser; used when the SSL (Security Socket Layer) is activated. The client may use this field in order to receive or send confidential information.

Time; date and time when a web object is requested by the web browser to the web

⁵Web logs are delimited text files as specified by RFC 2616, "Hypertext Transfer Protocol – HTTP/1.1" <http://www.rfc-editor.org/rfc/rfc2616.txt>

#	IP	Id	Access	Time	Method/URL/Protocol	Status	Bytes	Referer	Agent
1	165.182.168.101	-	-	16/06/2002:16:24:06	GET p1.htm HTTP/1.1	200	3821	out.htm	Mozilla/4.0 (MSIE 5.5; WinNT 5.1)
2	165.182.168.101	-	-	16/06/2002:16:24:10	GET A.gif HTTP/1.1	200	3766	p1.htm	Mozilla/4.0 (MSIE 5.5; WinNT 5.1)
3	165.182.168.101	-	-	16/06/2002:16:24:57	GET B.gif HTTP/1.1	200	2878	p1.htm	Mozilla/4.0 (MSIE 5.5; WinNT 5.1)
4	204.231.180.195	-	-	16/06/2002:16:32:06	GET p3.htm HTTP/1.1	304	0	-	Mozilla/4.0 (MSIE 6.0; Win98)
5	204.231.180.195	-	-	16/06/2002:16:32:20	GET C.gif HTTP/1.1	304	0	-	Mozilla/4.0 (MSIE 6.0; Win98)
6	204.231.180.195	-	-	16/06/2002:16:34:10	GET p1.htm HTTP/1.1	200	3821	p3.htm	Mozilla/4.0 (MSIE 6.0; Win98)
7	204.231.180.195	-	-	16/06/2002:16:34:31	GET A.gif HTTP/1.1	200	3766	p1.htm	Mozilla/4.0 (MSIE 6.0; Win98)
8	204.231.180.195	-	-	16/06/2002:16:34:53	GET B.gif HTTP/1.1	200	2878	p1.htm	Mozilla/4.0 (MSIE 6.0; Win98)
9	204.231.180.195	-	-	16/06/2002:16:38:40	GET p2.htm HTTP/1.1	200	2960	p1.htm	Mozilla/4.0 (MSIE 6.0; Win98)
10	165.182.168.101	-	-	16/06/2002:16:39:02	GET p1.htm HTTP/1.1	200	3821	out.htm	Mozilla/4.0 (MSIE 5.01; WinNT 5.1)
11	165.182.168.101	-	-	16/06/2002:16:39:15	GET A.gif HTTP/1.1	200	3766	p1.htm	Mozilla/4.0 (MSIE 5.01; WinNT 5.1)
12	165.182.168.101	-	-	16/06/2002:16:39:45	GET B.gif HTTP/1.1	200	2878	p1.htm	Mozilla/4.0 (MSIE 5.01; WinNT 5.1)
13	165.182.168.101	-	-	16/06/2002:16:39:58	GET p2.htm HTTP/1.1	200	2960	p1.htm	Mozilla/4.0 (MSIE 5.01; WinNT 5.1)
14	165.182.168.101	-	-	16/06/2002:16:42:03	GET p3.htm HTTP/1.1	200	4036	p2.htm	Mozilla/4.0 (MSIE 5.01; WinNT 5.1)
15	165.182.168.101	-	-	16/06/2002:16:42:07	GET p2.htm HTTP/1.1	200	2960	p1.htm	Mozilla/4.0 (MSIE 5.5; WinNT 5.1)
16	165.182.168.101	-	-	16/06/2002:16:42:08	GET C.gif HTTP/1.1	200	3423	p2.htm	Mozilla/4.0 (MSIE 5.01; WinNT 5.1)
17	204.231.180.195	-	-	16/06/2002:17:34:20	GET p3.htm HTTP/1.1	200	2342	out.htm	Mozilla/4.0 (MSIE 6.0; Win98)
18	204.231.180.195	-	-	16/06/2002:17:34:48	GET C.gif HTTP/1.1	200	3423	p2.htm	Mozilla/4.0 (MSIE 6.0; Win98)
19	204.231.180.195	-	-	16/06/2002:17:35:45	GET p4.htm HTTP/1.1	200	3523	p3.htm	Mozilla/4.0 (MSIE 6.0; Win98)
20	204.231.180.195	-	-	16/06/2002:17:35:56	GET D.gif HTTP/1.1	200	3231	p4.htm	Mozilla/4.0 (MSIE 6.0; Win98)
21	204.231.180.195	-	-	16/06/2002:17:36:06	GET E.gif HTTP/1.1	404	0	p4.htm	Mozilla/4.0 (MSIE 6.0; Win98)

Figure 2.2: A typical web log file

web server.

Request; an object requested by the browser.

Status; an integer code which shows the request status.

Bytes; the number of bytes returned in the request.

Referrer; a text string sent by the client that indicates the original source of a request or link.

User-Agent; the name and version of the client software making the request.

The log file shown in Fig. 2.2 contains the registers by their order of arrival. The illustration also shows the standard pieces of information stored in web log registers. The user's identification is not stored.

To analyze a user browsing behavior, it is necessary to reconstruct his or her real session. This is a very complex task due to noise contamination problems such as:

- Proxy and Firewall. If an institution uses proxies or firewalls, the real IP address

is masked by using the proxy or firewall external IP. The web log files will then contain large numbers of registers which originate from the same IP address.

- Web asynchronism. Usually, the user to a web site has no a priori identification, making it impossible to determine who owns a session. Identification methods like cookies or session reconstruction techniques are needed.
- Web crawlers. They are known as *spider robots* ⁶ used by search engines such as **Google** or **Yahoo!**, to collect web pages. In this case the web log will contain the requests of the robot so it will be necessary to identify these requests and eliminate them from the log as “artificial sessions”.

2.2.1 Session reconstruction process

The process of segmenting user activities into individual user sessions is called **sessionization** [63]. It is based on web log registers and due to the problems mentioned above, the process is not error free [209]. Sessionization usually assumes that each session has a maximum time duration. It is not always possible to identify user behaviour for if the page is in the browser cache and the user returns to it during the same session (back bottom in the browser), it would not be registered in the web logs. Thus invasive schemes such as sending another application to the browser have been proposed to capture user browsing accurately[21, 63]. However, this scheme can be easily be avoided by the user.

Many authors [21, 63, 163] have proposed using heuristics to reconstruct sessions from web logs. In essence, the idea is to create subsets of user visits as web log registers and apply techniques linking them together as one session during a certain period.

The purpose of session reconstruction is to find real user sessions, i.e., pages visited by a physical human being. In that sense, whatever the strategy chosen to

⁶<http://www.robotstxt.org/wc/robots.html>

discover real sessions, it must satisfy two essential criteria: that the activities can be grouped together; and that they belong to the same visit and to the same group.

There are several sessionization techniques that can be grouped in two major strategies: *proactive* and *reactive* [209].

Proactive strategies identify users with methods like cookies. When a user visits a web site for the first time, a cookie is sent to the browser. When the web site is revisited, the browser shows the cookie content to the web server, which automatically identifies it. The method has some technical problems and furthermore can jeopardize the user's privacy. First, if the site is revisited after several hours, the session will have a large duration, being the best way to classify it as another session. Secondly, some cookie features run foul of national or international data protection policies, for example those of the European Union [209]. Finally, the cookies can be easily detected and deactivated by the user.

Reactive strategies are noninvasive with respect to privacy and they make use of information contained in the web log files only; and build up user profiles through registers based on reconstructed sessions.

In general, web sites use reactive strategies rather than implementing identification mechanisms. These can, in turn, be classified into two main groups [23, 22, 63]:

- **Navigation Oriented Heuristics:** assumes that the user reaches pages through hyperlinks found in other pages. If a page request is unreachable through pages previously visited by the user, a new session is initiated.
- **Time Oriented Heuristics:** sets a maximum time duration, possibly around 30 minutes for the entire session [45]. Based on this value a specific session's transactions can be identified as belonging to the session by using program filters.

A first step in session reconstruction is to select only the relevant registers, usually those that have a direct relation to the visited pages, and eliminating registers which

refer to other objects, like pictures, sounds, videos or those describing status code errors.

By applying this procedure to the registers shown in Fig. 2.2, only a subset will then proceed to the next step, as shown in Fig. 2.3.

Web logs are contained in tabbed text-based files so that any programming language that can process streams, (like Perl, C, awk, etc.), can be used to group the registers by IP and agents, as shown in the left side of Fig. 2.3.

The second step categorises each register's group by time stamp. So the registers are selected from a time window of approximately 30 minutes and are grouped together into sessions, as shown in the right hand side of Fig. 2.3.

IP	Agent	Date	IP	Agent	Date	Sess
165.182.168.101	MSIE 5.01 16-Jun-02 16:39:02	165.182.168.101	MSIE 5.01	16-Jun-02 16:39:02	1
165.182.168.101	MSIE 5.01 16-Jun-02 16:39:58	165.182.168.101	MSIE 5.01	16-Jun-02 16:39:58	1
165.182.168.101	MSIE 5.01 16-Jun-02 16:42:03	165.182.168.101	MSIE 5.01	16-Jun-02 16:42:03	1
165.182.168.101	MSIE 5.5 16-Jun-02 16:24:06	165.182.168.101	MSIE 5.5	16-Jun-02 16:24:06	2
165.182.168.101	MSIE 5.5 16-Jun-02 16:26:05	165.182.168.101	MSIE 5.5	16-Jun-02 16:26:05	2
165.182.168.101	MSIE 5.5 16-Jun-02 16:42:07	165.182.168.101	MSIE 5.5	16-Jun-02 16:42:07	2
165.182.168.101	MSIE 5.5 16-Jun-02 16:58:03	204.231.180.195	MSIE 6.0	16-Jun-02 16:32:06	3
204.231.180.195	MSIE 6.0 16-Jun-02 16:32:06	204.231.180.195	MSIE 6.0	16-Jun-02 16:34:10	3
204.231.180.195	MSIE 6.0 16-Jun-02 16:34:10	204.231.180.195	MSIE 6.0	16-Jun-02 16:38:40	3
204.231.180.195	MSIE 6.0 16-Jun-02 16:38:40	204.231.180.195	MSIE 6.0	16-Jun-02 17:34:20	4
204.231.180.195	MSIE 6.0 16-Jun-02 17:34:20	204.231.180.195	MSIE 6.0	16-Jun-02 17:35:45	4
204.231.180.195	MSIE 6.0 16-Jun-02 17:35:45				

Figure 2.3: Sessionization process using a time based heuristic

A prior step is to identify irregular or deviant sessions, for example, registers that do not belong to human users but to web robots or spiders. This cleaning step can be implemented by reviewing the agent parameter, for if the user is a robot, the agent usually reveals that information. However, if the robot does not identify itself, we have the firewall situation, i.e., a long session performed by one user. But as noted, it is possible to apply a filter to eliminate long sessions during the sessionization process.

Sometimes the action of crawlers became a huge problem for analyzing web logs, because they can generate thousand of unreal visits to a given site in a short period. This problem especially complicates web-based systems that provide online navigation

recommendations, given a big motivation for the early crawlers' detection for filtering these sessions [78]. There are both official⁷ and unofficial⁸ lists of crawlers maintained for identification.

Sorting and grouping processes use considerable resources and programming may be less efficient than commercial tools. Hence alternative tools, such as a relational database engines that use tables when loading registers and objects like indexes, speed the grouping and sorting process.

2.2.2 Finding real sessions

The sessionization process cleans and prepares web logs for identifying user sessions to a web site. However, more conditions must be applied to identify a “real session”, i.e., the sequence of pages visited by a (non web crawler) web site user.

Let L be the set of web log registers and $R = \{r_1, \dots, r_n\}$ the initial set of the real sessions, extracted from L through the sessionization process.

The minimal conditions [209] for r_i in order to be a real session are:

1. To be composed by the objects requested during the session and ordered by time stamp. Then,

$$\forall r_i \in R, \forall j = 2, \dots, \text{length}(r_i) \quad r_{i,j}.\text{timestamp} > r_{i,j-1}.\text{timestamp}$$

2. Only objects requested in L can appear in R , i.e., $\bigcup_{r_i \in R} (\bigcup_{j=1}^{\text{length}(r_i)} r_{i,j}) = L$.

3. Each request in L belongs to exactly one session of R , i.e.,

$$\forall r_i \in R, \forall j = 1, \dots, \text{length}(r_i) : \nexists i' \neq i, j' / r_{i,j} = r_{i',j'}$$

These properties ensure that R partitions L in an order-preserving way.

⁷<http://www.robotstxt.org/>

⁸<http://www.psychedelix.com/agents/index.shtml>, <http://www.pgts.com.au/pgtsj/pgtsj0208d.html>

By applying the above conditions, the number of wrongly identified sessions is reduced. However, there will always be misclassified sessions. This is why a further additional postprocessing task is required to distinguish wrong behavior from the real sessions. Again, this reduces the number of poorly identified sessions but can not totally eliminate them [22].

2.3 The information contained in a web page

Objects like pictures, movies, sounds, page links and free text can be found in most web pages and are closely linked to user preferences. By understanding which specific objects the user is looking for allows an improvement in web site content and makes the site more attractive for the eventual users. In page content analysis, two objects receive special attention: free text and links to others pages. These two types of elements are processed in different ways for data mining.

2.3.1 Web page content

Web texts can be organized as free text, semi-structured as HTML, or fully structured as tables or databases [176]. Web user preferences can be represented using these categories [154]. For example, understanding which words or concepts are more important to the user may help to create new content for web pages [228]. Beyond analyzing the words contained in the web pages, we are also interested in extracting concepts from hyperlinks [47], which are related to the information that the user is searching for.

In the area of Information Retrieval (IR), the documents are traditionally represented using the **vector space model** [1, 14, 198]. A first step is word representation based on a tokenized process using simple syntactic rules, with tokens stemmed to canonical form (e.g., ‘writing’ to ‘write’; ‘are’, ‘did’, ‘were’ to ‘be’). Next, a document is represented in a vector space.

Let R be the number of different words in the entire set of documents and Q be the number of documents. A vectorial representation of the web site would be a matrix M of dimension $R \times Q$ with:

$$M = (m_{ij}) \quad i = 1, \dots, R \quad \text{and} \quad j = 1, \dots, Q \quad (2.1)$$

where m_{ij} is the weight of word i in document j . The weight must capture the fact that a word can be more important than another [196]. For instance, if the word i appears in n_i documents, the expression $\frac{n_i}{Q}$ gives a sense of importance in the complete set. The “inverse document frequency”, $IDF = \log(\frac{Q}{n_i})$, is used as a weight.

A variant of this is to apply a factor to the IDF. The resulting expression is known as TFIDF (**T**erm **F**requency / **I**nverse **D**ocument **F**requency), and is given by,

$$m_{ij} = f_{ij} * \log(\frac{Q}{n_i}) \quad (2.2)$$

where f_{ij} is the number of occurrences of word i in the document j .

In the vectorial representation, a document is a column of the matrix M , containing R elements. The column content can be extracted and used as inputs to data mining algorithms [192].

It is worth recalling that a web page is not a normal text document because it mixes human “readable” text with HTML tags which is why it needs a preprocessing and cleaning operation before using the vector space model (see Fig. 2.4).

Since the representation of the hypertext depends on the expected application, it is necessary to first identify what elements are necessary to model. As seen in Fig. 2.4, a cleaning task removes the HTML tags. However, the text in between two tags is sometimes related to content, e.g., the $< title >$ tag shows the main page theme and

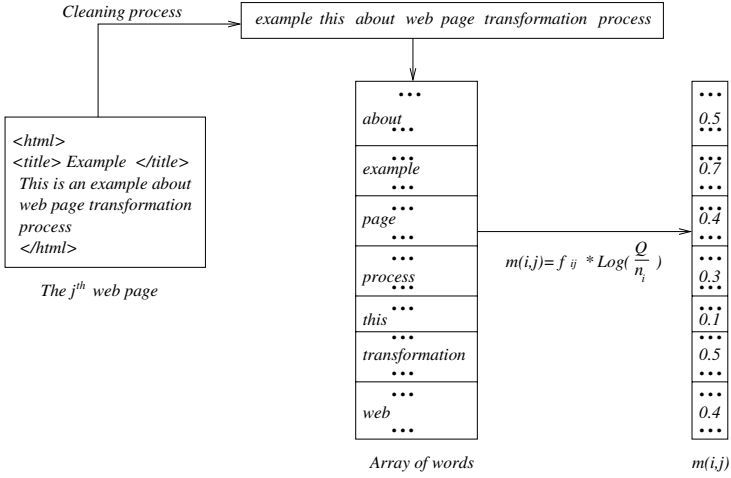


Figure 2.4: A web page represented in the vector space model

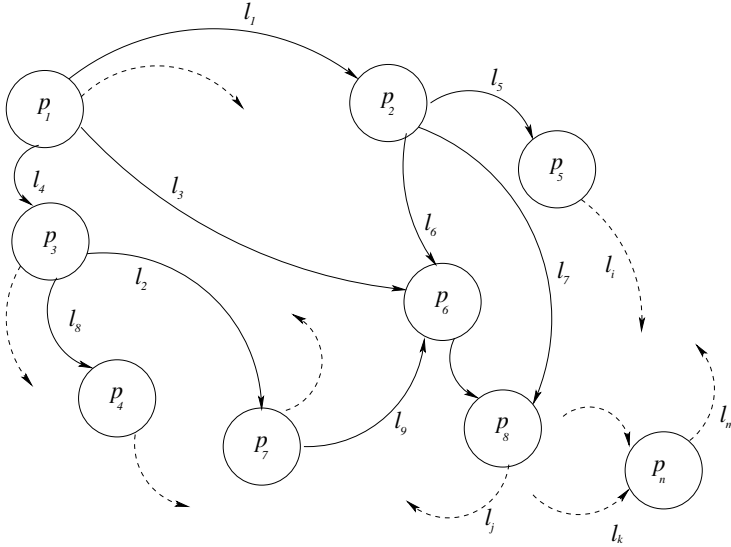
could be incorporated in the final expression of the hypertext vector representation with a different weight, to get some idea about the user text preferences [242].

2.3.2 Web page links

When a web page contains a hyperlink pointing to another web page, usually these two pages become related by content. If a set of web pages have hyperlinks among them, these links create a community made up of common information [97], as shown in Fig. 2.5.

By construction, Fig. 2.5 can be seen as a directed graph [131], where node pages point to other node pages, depending on the information contained in those web pages. It is reasonable to determine that different pages have different relevances within the community.

A simple web page classification sets out which pages contain the more relevant information. Based on the hyperlinks structure, it is possible to distinguish two kinds



2.4 Summary

Three types of web data have special significance for discovering new patterns about web user behavior: web logs, web pages and the web hyperlinks structure. Because some of this data is semi-structured, often unlabeled and with high levels of noise, they must be preprocessed and cleaned before being used as input for web mining tools.

The sessionization process, by using web logs, allows for the reconstruction of the original user session. However, this is only an approximation as user identification is not completely possible, as partial session identifiers (IP, agents, etc.) have to be used. This explains the need for *a posteriori* cleaning operations for unreal sessions, like those initiated by crawlers. Again, this reduces the noise, but does not eliminate it completely.

Web pages need to be transformed into feature vectors, and a simple way to do this is by using the vector space model. However, it is important to remember that the cleaning process must identify important words for further pattern discovery tasks, particularly those lying between two tags.

The web hyperlinks structure differentiates between pages of importance and unimportance. Further, a picture of a web community emerges when these are classified into authoritative and hub pages. Pages can be given weight relative to their importance to the entire web community.

To conclude, interesting patterns about the user behavior can be constructed from web data and about web networks around a given set of topics. Trustworthy patterns, discovered through web mining techniques, depend on satisfactory data; and this in turn depends on (after strong cleaning and preprocessing tasks for reducing unnecessary noise) the quality of inputs from the web.

Chapter 3

Knowledge discovery from web data

*Knowledge is not skill. Knowledge plus
ten thousand times is skill.*

Shinichi Suzuki

Institutions store huge volumes of data obtained from their operational systems like sales, inventory, etc. [113]. Managers are aware that data is the key to the success of their companies, but often do not have real access to this information due to software usage complexity, system architectures, or simply because they do not understand technology.

When an operational system produces raw data, it is rarely used directly in the raw form; this raw data needs to be transformed in order to provide the right information for the business user, also called “*the end-user*”. The greater the amount of raw data, the more difficult it is for conventional tools like SQL queries, **OnLine Analytical Processing** (OLAP) tools or Decision Support Systems (DSS) [53, 58] to process it. The main focus of the “Knowledge Discovery in Data Bases” (KDD) process [84, 85, 86] is to automate data processing, which allows users to become more efficient in their data analysis tasks and to find facts and relations among data.

The key characteristics of web data - its volume, controlled noise and hidden patterns - make the KDD process very suitable to extract information from web data [134]. The information extraction and the use of the extracted information as knowledge is a non-trivial task since it deals with a huge collection of heterogeneous, unlabelled, distributed, time variant, semi-structured and multi dimensional data [176]. Information extraction from web sources requires new models (both for using and storing the data efficiently) as well as the development of new algorithms called web mining algorithms [136].

Knowledge on web site user behavior can, if properly interpreted, be used for improving the structure and content of the site by (off line) modifications. In addition, this knowledge also can be used for the development of sophisticated systems and for the provision of proposals for better navigations [180, 183, 243] through online recommendations about which pages the web users should visit to find information about they seek.

3.1 Overview

KDD is defined as *“the process of nontrivial extraction of information from data, information that is implicitly present in that data, previously unknown and potentially useful for the user”* [94] and is a generic methodology for finding information in a large collection of data.

Let F be the set of data from the operational sources and c a measure to compare how similar two elements are of F . A pattern F_s is an expression S such that $F_s \subseteq F$. In this sense, knowledge discovery means finding a pattern sufficiently interesting for the user, and the measure c must be specified by the user.

Using the above definitions, we can consider that “knowledge discovery” is a result of the application of c to F , with the product being the resulting pattern. A measure needs a quantitative expression that reflects its degree of importance for the

user. Only interesting information patterns can be transformed into knowledge and need to be new, non trivial and useful.

The KDD process considers a sequence of stages [249], as shown in Fig. 3.1. It is an iterative process, and it is possible to return to previous stages and adjust parameters or check the assumptions.

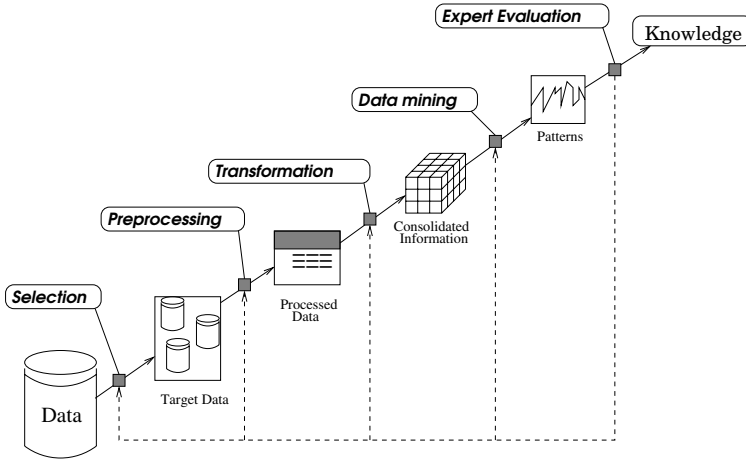


Figure 3.1: The stages of the KDD process.
(based on Fayyad et al. [86])

The process begins with the selection of the operational systems¹ from which the data will be collected. It is likely to encounter many data problems. For example, some of the registers may be wrong, because when the system was tested, fictitious or unreliable data were inserted and were never erased. Frequently, the operational systems have validation problems, i.e., they accept erroneous data from the users. For these reasons, a preprocessing stage is necessary before applying any tools to discover patterns from the data.

¹Computational systems that support the daily tasks in an institution, for example, the sales system in a shop.

The next KDD steps are the consolidation and the summarization of the information. A data base model such as the Data Warehouse and/or Data Mart architecture could be used in this stage to create an information repository [111, 128].

Application of non-conventional techniques, like SQL queries or OLAPs tools, are required to discover patterns. Data mining techniques [104] can reveal hidden information which can be formulated as predictive models. They will need to be validated by business experts.

The pattern discovery task is implemented by using “data mining” techniques [5, 25, 26, 76, 161]. Before this, a cleaning and preprocessing task must be performed over the data to be prospected.

The role of business experts is inextricably linked to the interpretation of the pattern analysis; they are able to judge the best degree of fit of data mining results, models, measures and algorithms - developed in the previous stages. This task is very important as it validates the patterns found and makes proposals about how to use them in order to improve the business. Frequently, an expert can perceive certain behavior but may not be able to demonstrate it empirically; so, in such cases, data mining could help validate or reject an hypothesis.

All the above mentioned stages are quite complex and each one can be a subject of study in itself. Notice that not all of the stages need to be incorporated in every specific KDD project and some of them can be more emphasized than others.

3.2 Data sources and cleaning

The identification of real data sources is an important step in the KDD process [84]. The use of irrelevant data often leads to analytic errors as well as adding noise to the final results.

In the past, operational systems used proprietary platforms to load data, usually

in formats that could be read only by using programs written in the original language. This approach may lead to difficulties especially when it is necessary to mix files or to communicate among systems [128].

The next step is the standardization of the file format. Often, the files were written using only ASCII characters and some simple meta-data² about the content. Many of these files are highly complex and the final data file structure requires non trivial meta-data to explain the content. In many cases, this meta-data was in the mind of the programmer and never written down, so if he/she leaves the institution, this important information goes with him/her.

The most frequent data problems, based on a literature review and practical cases [113, 128], are:

- **Data consistency.** Very frequently, the data has consistency problems, as the operational system has not correctly validated the data.
- **Data manipulation errors** which tend to occur when operational systems tests are undertaken or updated without correct planning.
- **Irrelevant Data.** Depending on the kind of data analysis to be performed, it is necessary to filter out irrelevant data.

To put it succinctly, “*garbage-in, garbage-out*”; incorrect data lead to incorrect results, so that correcting steps are mandatory for a successful data mining process. And what about the question of dealing with web data? In addition to these problems, web data is highly variable in type and format. Data types used in traditional operational systems are well known - integer, real, char and in some cases long char, date, etc. In the last decade, new data types have been incorporated to allow for the storage of sound, text, movies and images, even though information extraction tasks continued to focus on the previous data types.

²Information about the data

The web page content covers all the existing data types, by including tags inside of the HTML texts. The data has relatively high content variation, which depends on the frequency of page updates. Web hyperlink data have not changed in format from the beginning of the Web. However, it is not always easy to understand the content pointed at by a hyperlink. So also, web logs are regulated by their format. However, the web master is at liberty to decide which data will be included in the final format of a log register; for example, the parameter “agent” could be interesting and included in the log format for one web site but omitted in another. The problem with web logs is the huge amount of irrelevant data normally generated by web operations. Furthermore, some web logs lack the data which can help determine the user’s browsing behavior at the web site.

An important task of the KDD process is the transformation of data into feature vectors as inputs for data mining tools. The web data cleaning and preprocessing activities should store the results in an information repository which will allow statistical queries and feature vectors generation.

3.3 Data consolidation and information repositories

An institution’s information architecture can be simple or complex depending on the computing background of that institution, i.e., for how long its business has been supported by informatics systems. It could be constituted by heterogeneous operational systems, which in many cases may not work together.

How can we integrate the information from a simple MS Excel file with a database in Cobol? The answer is apparently straightforward - by systems integration. But, the implementation is likely to be very complex, as it should not disturb the normal functioning of the operational system while allowing a minimum integration between them.

Apart from integration, there are other difficulties involved. Supposing that all institution's operational systems have the same informatics platform, what would happen to the data storage conventions? There may be, for example, two sales systems in the same transnational company. In one, weights are measured in pounds and in the other in kilograms, so how will the data be combined. Any answer is likely to involve complex implementation challenges.

These situations become even more complex when the time variable is considered; i.e., the systems change over time modifying data structures, business rules, standards, etc., and it becomes in general more difficult to undertake integration and consolidation tasks. These tasks - integration and consolidation - should depend on a comprehensive strategy prior to storing the information in a repository.

The Data Staging Area (DSA) [128] is the location where data are consolidated and transformed into information. The DSA can be physically implemented using a computer in the network, from which the data extraction is undertaken from the operational systems and the data transformation is performed, based on the standard data format previously defined. This process is known as Extraction, Transformation and Loading (ETL).

Finally, the information repository will contain the results of the complete data process, i.e., cleaning, preprocessing, integration and consolidation. This consolidated information will be stored in a repository, available for queries.

The information repository will become the historic memory of the institution as it receives the results of processes executed periodically. It can be used to extrapolate future action based on past records, which make it a suitable source for data mining.

Of the different architectures, the data warehousing represents a methodology that guarantees success in the development of an information repository platform that supports the whole process requirements. This architecture has been successfully used in the construction of a **Webhouse** [128], i.e., a data warehouse for web data.

3.4 Data Mining

Data mining is the core of the KDD process and can be briefly described as the process of extracting patterns from a set of data. Data mining has been a very active field as, recently, the amount of data being produced has increased exponentially whilst the cost of storage devices, like hard disks, has decreased. Moreover, classical statistics techniques are inefficient when compared to data mining. For example, as a general characterisation, data mining algorithms are able to scale up well to large data sets, while traditional statistical algorithms execute in quadratic run-time.

Another significant difference is with respect to data focalization. When using data mining tools, the data is supposed to be independent of the algorithms to be used, while regarding traditional methods, the data is collected for a specific tool, as for example when a sample of a market is taken for a specific statistical algorithms that is employed. There are other characteristics that make data mining techniques more attractive for processing large amounts of data. The next section introduces the main ideas and techniques underlying the data mining.

3.4.1 Motivation

A case, from a large US supermarket, best explains the essence of data mining. From customer purchase behaviour, data mining tools searched for correlated products, that is, products linked to one another (e.g. bread \implies butter). Most findings seemed self evident, but some were puzzling as for the case of beer and diapers. Was this an error or was the data mining tools showing a hitherto hidden pattern? The results were then shown to a retail business expert. After examining the results he could see that the majority of these linked purchases were executed by men on Friday afternoons. But which market segment did they belong to - retired, adults, young adults, etc.? It was revealed that the market segment consisted of young men, married in the last three years with small children.

With this information, that young fathers go to the supermarket to buy diapers and beer on Friday afternoons, the supermarket decided to place the two related products side by side. It was assumed that a father would not forget one of these two products, as he would see both of them at the same time. This change produced a significant sale increase for both products.

The information model used as input to the data mining algorithms is key to pattern discovery. And the most obvious way to validate the discovered patterns is, as in the case above, to show them to business experts. Their knowledge and expertise helps modify the model.

3.4.2 Data Mining techniques

In this section, we present a short review of the fundamental concepts underlying the data mining techniques [25, 26, 170], showing the main algorithms and examples about their practical operations.

3.4.2.1 Association rules

The basic idea in association rules is to find significant correlations among a large data set. A typical example of this technique is the “purchasing analysis”, which uses customer buying habits to discover associations among purchased items. The discovery rules are formalized as *if* $\langle X \rangle$ *then* $\langle Y \rangle$ expressions.

The formal statement for the described situation is proposed in [6, 8] as follow. Let $I = \{i_1, \dots, i_m\}$ be a set of items and $T = \{t_1, \dots, t_n\}$ be a set of transactions, where t_i contains a group of items from I . Let $X \subseteq I$ be a group of items from I , a transaction t_i is said to contain X if $X \subseteq t_i$. An association rule is an implication of the form $X \Rightarrow Y$, where $Y \subseteq I$, $X \cap Y = \phi$.

The rule $X \Rightarrow Y$ holds for the transactions set T with support α and confidence

β , where α is the percentage of transactions in T that contain X and Y and β is the percentage of transactions in T that contain $X \cup Y$. This process can be generalized with the multidimensional association, whose general form is $X_1, \dots, X_n \Rightarrow Y$.

Depending on the number of transactions and items, the explained process could require a considerable amount of computer resources. Then a selective criterion is required. A rule is considered interesting if satisfying a minimum α and β . As example, consider the diaper case introduced in section 3.4.1. In the supermarket case, an association rule could be in the form shown in the expression 3.1:

$$beer \Rightarrow diapers \text{ [support} = 5\%, \text{ confidence} = 53\%]. \quad (3.1)$$

The rule states that “the persons who purchase beer tend to buy diapers at the same time”. The support measure means that 5% of all buying transactions show this combination of products, and the confidence measure states that 53% of the customers who purchase beer also buy diapers.

3.4.2.2 Classification

Given a set of objects, classification techniques sort objects into distinct categories or classes [80]. This involves two steps: learning and classification. During learning, a random data sample is used to test a proposed model, by comparing the predicted classes with the real classes to which the objects belong. The training is iterative and stops when the rate of correct classifications is superior to a certain threshold.

3.4.2.3 Clustering

Clustering is the process of grouping objects with similar characteristics [107]. In supervised learning, as described above, the class label is known *a priori*. The main idea of clustering is to identify classes of objects (clusters) that are homogeneous

within each class and heterogeneous between different classes [187]. Clustering is usually synonym to unsupervised learning.

Let Ω be a set of m vectors $\omega_i \in \mathfrak{R}^n$, with $i = 1, \dots, m$. The goal is to partition Ω into L groups, where C_l denotes the l^{th} cluster, $l = 1, \dots, L$. Then, $\omega_i \in C_l$ means that ω_i has greater similarity to elements in cluster C_l than to elements belonging to any other cluster. Clustering requires a similarity measure [116], $\zeta(\omega_p, \omega_q)$ by which it compares two vectors from Ω .

For each $\omega_i \in C_l$, its influence toward others vectors in the cluster, also called “influence in the neighborhood”, is calculated by a *Kernel Function* [168] $\kappa_B^{\omega_i} : \Omega \rightarrow \mathfrak{R}$, where B is the type of the function, for instance,

$$\kappa_{Gauss}^{\omega_i}(\omega_j) = e^{-\frac{\zeta(\omega_i, \omega_j)^2}{2\sigma^2}}, \quad (3.2)$$

is a Gaussian kernel function with $\omega_j \in C_l$ and smoothness σ , which shows the influence of a vector ω_j at a vector ω_i .

The density of a vector $\omega \in C_l$ is calculated by using a *Density Function* $D_B^{C_l}(\omega)$, as the sum of influences of all vectors $\omega_i \in C_l$, i.e.,

$$D_B^{C_l}(\omega) = \sum_{\omega_i \in C_l} \kappa_B^{\omega_i}(\omega) \quad (3.3)$$

The determination of the number of clusters depends on the method used. The most relevant and frequently used clustering techniques are:

Partition Clustering. Divide n objects into k groups called partitions. Let $X = \{x_1, \dots, x_n\}$ be the set of objects and $P = \{p_1, \dots, p_k\}$ the desired partitions, with $k \leq n$. A partition p_i is a subset of X such that:

- $p_i \neq \phi \quad \forall \quad i = 1, \dots, k.$
- $X = \cup_{i=1}^k p_i.$

- $p_i \cap p_j = \phi \quad \forall i \neq j$

Hierarchical Clustering. The main idea is to build a hierarchical decomposition of a data set using either an agglomerative or a divisive method. The first approach begins defining each point as a cluster. Then, similar clusters are merged until a terminal condition is met. In contrast, the second approach assumes that initially all objects belong to the same cluster and then split it up, in a top-down manner, until the terminal condition is satisfied.

Density Based Clustering. It employs the density notion used in physics. More exactly, the clusters are discovered when a neighborhood exceeds a density threshold. Let $X = \{x_1, \dots, x_n\}$ be the data set, and $C = \{c_1^i, \dots, c_m^i\}$ the cluster set in the i^{th} iteration. Let δ be the density threshold and $card(c_k)$ the cardinality of data in cluster c_k . A point x_j belongs to c_k^i if $D(c_k^i, x_j) \leq R$, where D is a distance measure, like the Euclidean one, and R a radius. The process stops when $\forall c_k^i \in C, card(c_k^i) \leq \delta$, otherwise the cluster centroids are redefined.

Cluster identification is difficult because it depends on subjective factors [216]. There are two approaches which can help estimate the point at which a group becomes a cluster [107]. Firstly, the kernel function considers the influence of neighborhood data points. Secondly, the density function is the sum of the influence of all data points. In both cases, the cluster identification needs a parameter whose value depends on the specific problem under investigation.

Again, as a concluding remark, cluster interpretation depends on subjective judgements and that is why it is desirable to have expert help when clustering the data [216].

3.5 Tools for mining data

Data mining techniques have to be flexible; the algorithms always require modifications or adaptations because both information and pattern extraction tasks tend to be application dependent. This section provides a short survey on the most used data mining techniques, which also illustrates their basic operation and main restrictions.

3.5.1 Artificial Neural Networks (ANN)

ANN represents a mathematical model for the operation of biological neurons from brain and, like biological neural structures, ANNs are usually organized in layers. A simple model for a neuron is shown in Fig. 3.2. The neuron is modelled as an activation function that receives stimulus from others neurons, represented by inputs with associate weights.

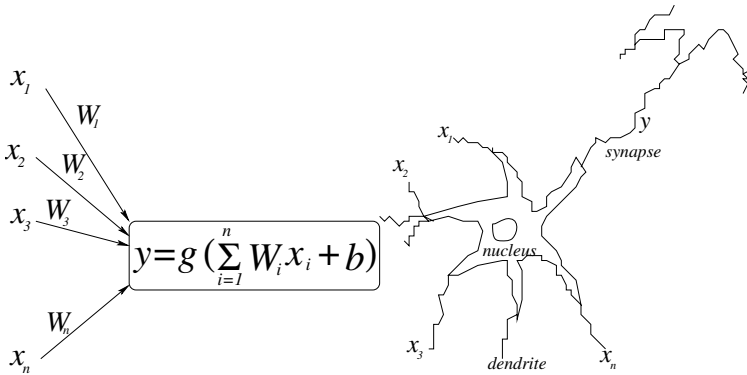


Figure 3.2: A biological neuron representation

The expression

$$y = g(\sum_i W_i x_i + b), \quad (3.4)$$

is the most widely used model of an artificial neuron. A “perceptron” network is a one-layer ANN, and it was first used by Rosenblatt in 1957 [194] for basic classification tasks. In the expression (3.4), the W_i values are known as “neuron weights” and they can be identified following the minimum quadratic error criterion, i.e., the total error is defined as

$$E = \sum_i (y_i - g(\sum_k W_k x_{ik}))^2. \quad (3.5)$$

and then the local minimum is found when

$$\frac{\partial E}{\partial W_j} = 0. \quad (3.6)$$

The values for W_j are not directly found from solving equation 3.6. However, based on the “gradient descent rule”, an approximation is found in successive iterations.

For simplicity, if f is a scalar function $f(W) : \Re \rightarrow \Re$, then the weights are updated as given below:

$$W \leftarrow W - \eta \frac{\partial f(W)}{\partial W}. \quad (3.7)$$

The parameter η is known as the “learning rate”. Then the expression that minimize the quadratic error is

$$W \leftarrow W - \eta \frac{\partial E}{\partial W}, \quad (3.8)$$

where

$$\begin{aligned}
 \frac{\partial E}{\partial W_j} &= \sum_i 2(y_i - g(\sum_k W_k x_{ik})) \left(-\frac{\partial g(\sum_k W_k x_{ik})}{\partial W_j} \right) \\
 &= \sum_i -2(y_i - g(\sum_k W_k x_{ik})) g'(\sum_k W_k x_{ik}) \frac{\partial(\sum_k W_k x_{ik})}{\partial W_j} \\
 &= -2 \sum_i (y_i - g(\sum_k W_k x_{ik})) g'(\sum_k W_k x_{ik}) x_{ij}.
 \end{aligned} \tag{3.9}$$

Different non-linear functions can be used to represent the activation function g , the most used one being the sigmoid function:

$$g(x) = \frac{1}{1 + e^{-x}}, \tag{3.10}$$

which has an interesting characteristic:

$$g'(x) = g(x)(1 - g(x)). \tag{3.11}$$

Then, equation 3.9 can be rewritten as:

$$\begin{aligned}
 \frac{\partial E}{\partial W_j} &= -2 \sum_i (y_i - g(\sum_k W_k x_{ik})) g(\sum_k W_k x_{ik}) (1 - g(\sum_k W_k x_{ik})) x_{ij} \\
 &= -2 \sum_i (y_i - g_i) g_i (1 - g_i) x_{ij}
 \end{aligned} \tag{3.12}$$

where $g_i = g(\sum_k W_k x_{ik})$.

Finally, the expression (3.7) becomes:

$$W_j \leftarrow W_j + \eta \sum_i (y_i - g_i) g_i (1 - g_i) x_{ij}, \tag{3.13}$$

which is known as the “neuron training” formula.

The simple one-layer ANN was used to introduce the more complex multi-layer ANN, as shown in Fig. 3.3. This ANN has two input neurons ($N_{INPUT} = 2$), one

hidden layer with three neurons ($N_{HIDDEN} = 3$) and one output neuron. This model can become more complex, with more hidden layers, more inputs or outputs neurons.

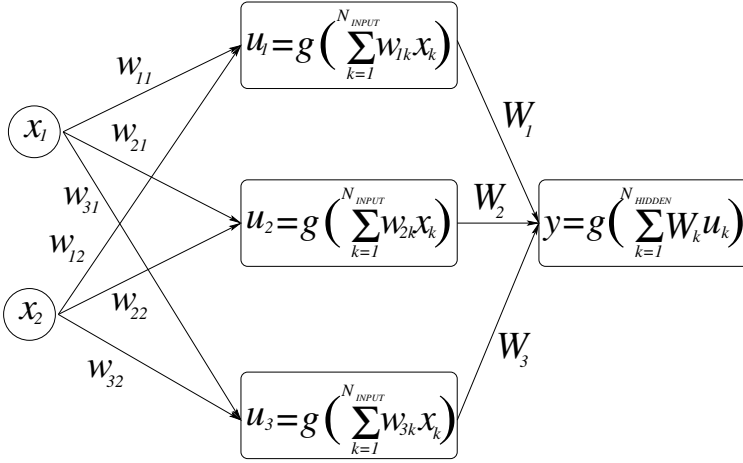


Figure 3.3: A multi-layer Artificial Neural Network

In order to train a multi-layer ANN, it is necessary to minimize the network error, beginning with the output neurons, which implement the following relation:

$$y = g\left(\sum_j W_j g\left(\sum_k w_{jk} x_k\right)\right), \quad (3.14)$$

and, again, we need to find the set of weights W_j and w_{jk} to minimize the error function:

$$E = \sum_i (y_i - g(\sum_j W_j g(\sum_k w_{jk} x_k))), \quad (3.15)$$

using the gradient descent rule.

This training method is known as the “Back-Propagation Algorithm” and its

convergence towards the minimum error is not guaranteed.

Multi-layer ANNs are mainly used in:

- **Classification.** By training an ANN, the output can be used as a feature vector classifier, for example, a customer risk classification at a Bank. The input layer receives a n -dimensional vector with a person's economic circumstances and in the output layer, she/he can be classified as “risky” or “not-risky”, depending on the neuron result being closer to “1” or “0”.
- **Prediction.** Given a set of training examples, an ANN can be trained to represent the approximate function on which to model a situation. When presented with new input examples, the ANN provides the best prediction based on what was learned using the training set of examples.

3.5.2 Self-Organizing Feature Maps (SOFMs)

A SOFM is a vector quantization process that takes a set of vectors as high-dimensional inputs and maps them into an ordered sequence. The SOFM maps from the input data space \mathfrak{R}^n onto a regular two-dimensional array of nodes or neurons. The output lattice can be rectangular or hexagonal. Each neuron is an n -dimensional vector $m_i \in \mathfrak{R}^n$, whose components are the synaptic weights. By construction, all the neurons receive the same input at a given moment of time.

A SOFM is a “*nonlinear projection of the probability density function of the high-dimensional input data onto the bi-dimensional display*” [135]. Let $x \in \mathfrak{R}^n$ be an input data vector. The idea of this learning process is to present x to the network and, by using a metric, to determine the most similar neuron (center of excitation, winner neuron).

The winner neuron, m_c , is defined as the “best-matching” in the whole network,

$$\|x - m_c\| = \min\{\|x - m_i\|\} \text{ or } c = \arg \min\{\|x - m_i\|\} \quad \forall \quad i = 1, \dots, n. \quad (3.16)$$

The initial values $m_i(0)$ are randomly set. Then the neighbor nodes of the winner neuron are activated to “learn” the same sample. The weight update rule is:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)], \quad (3.17)$$

where t is an integer and represents the iteration step, and h_{ci} is called “neighborhood kernel”. It is a function defined over the output lattice nodes, e.g. $h_{ci}(t) = h(\|r_c - r_i\|, t)$, where $r_c, r_i \in \mathfrak{R}$ are the winner neuron position and the i^{th} neuron in position in the array. The function h_{ci} is such that when $\|r_c - r_i\|$ increases, $h_{ci} \rightarrow 0$.

Depending on which points are considered in h_{ci} , the notion of neighborhood changes, providing diverse topologies:

- Open topology; maintains the two-dimensional condition in the array.
- Tape topology; the nearest neighbors of the neurons on the right edge are those situated on the left edge.
- Thoroidal topology; the closest neurons to the ones on the superior edge are located on the inferior edge maintaining the lateral neighborhood as in the tape topology.

The three topologies are shown in Fig. 3.4.

There are several definitions for h_{ci} that can be used, depending on the data to be mapped and other variables such as the distance covered. However, the idea is that the function should have a decreasing impact on the training of the neurons more distant from the winner.

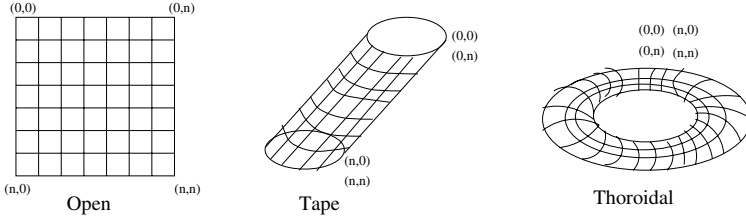


Figure 3.4: SOFM topologies

A widely used neighborhood kernel, based on a Gaussian function, the following illustrates:

$$h_{ci} = \alpha(t) \cdot e^{-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}}, \quad (3.18)$$

where $\alpha(t)$ is a scalar value called “learning rate” and $\sigma(t)$ defines the width of the kernel. Both $\alpha(t)$ and $\sigma(t)$ are decreasing functions in time.

3.5.3 K-means

This algorithm divides a set of data into a predetermined number of clusters [105]. The main idea is to assign each vector to a set of given cluster centroids and then update the centroids given the previously established assignment. This procedure is repeated iteratively until a certain stopping criterion is fulfilled.

The number of clusters to be found, k , is a required input value for the K-means algorithm. A set of k vectors are selected from the original data as initial centroids. This initial selection can be performed randomly. The clustering process executes the following iterative algorithm.

Given c_1^j, \dots, c_k^j as cluster centroids in iteration j , we compute $c_1^{j+1}, \dots, c_k^{j+1}$ as cluster centroids in iteration $j + 1$, according to the following steps:

1. Cluster assignment. For each vector in the training set, determine the cluster to which it belongs, using a similarity measure to compare the vectors.
2. Cluster centroid update. Let $V_l^j = \{v_1, \dots, v_{q_l^j}\}$ be the set of q_l^j vectors associated to centroid c_l^j , with $l = 1, \dots, k$. The next centroid is determined as c_l^{j+1} , the mean of V_l^j , i.e., $v_i \in V_l^j / \max\{\sum_{j=1}^{q_l^j} sm(v_i, v_j)\}, i \neq j$, where sm is the similarity measure.
3. Stop when $c_l^{j+1} \approx c_l^j$.

The K-means algorithm is shown in its first iteration in Fig. 3.5. Here, the initial centroids are selected randomly. Next, the new centroids are calculated as described above and are represented in Fig. 3.5 as “squares”.

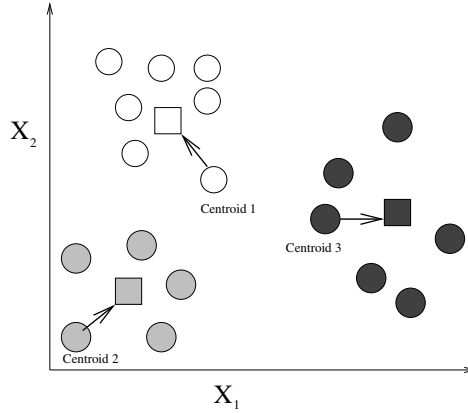


Figure 3.5: K-means after the first iteration

3.5.4 Decisions trees

A decision tree [38] is a data structure, in which the search among the tree-like data structure that classifies a set of data is done using some selection criteria for each

branch. To understand how decision trees are created, consider the data in Table 4.1., which corresponds to scarf purchases for different days and weather conditions at the beginning of the winter. The table shows only a part of all transactions carried out by a shop, which has been chosen for didactic purposes only.

Buy?	Weather	Buyer	Age
yes	Cold	man	<15
no	Cold	man	15-20
yes	Cold	man	> 50
yes	Cold	woman	<15
yes	Cold	woman	21-30
no	Rainy	man	31-40
yes	Rainy	man	40-50
yes	Rainy	woman	>50
no	Rainy	woman	21-30
no	Rainy	woman	15-20
no	Sunny	man	<15
no	Sunny	man	15-20
no	Sunny	woman	21-30
no	Sunny	woman	31-40
no	Sunny	woman	40-50
yes	Warm	man	>50
no	Warm	man	40-50
yes	Warm	woman	>50
no	Warm	woman	21-30
no	Windy	man	21-30
no	Windy	man	15-20
yes	Windy	woman	31-40
yes	Windy	woman	>50
no	Windy	woman	15-20

Table 3.1: Scarf's purchase behavior used to create a decision tree

Using only the data from Table 4.1, Fig. 3.6 shows the structure of the corresponding decision tree. Some rules can be induced from this tree and used to classify and predict future purchasing behavior, for example:

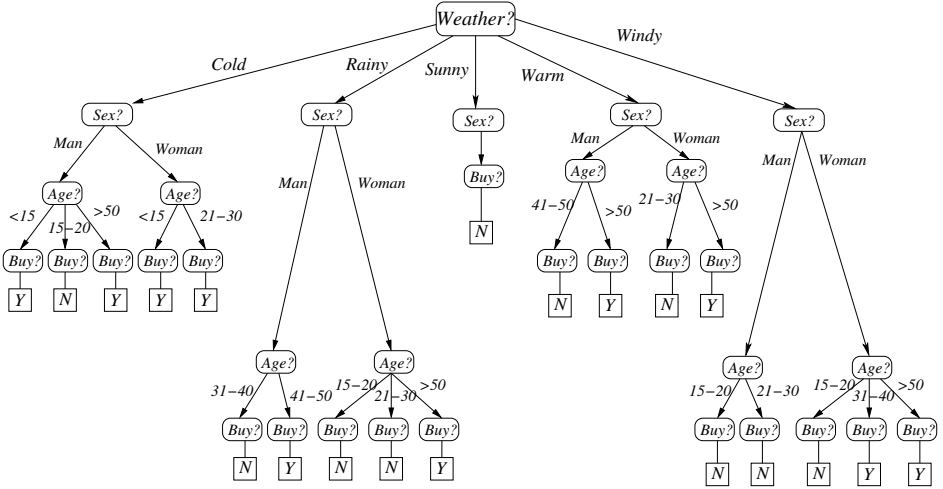


Figure 3.6: A decision tree for scarf purchases

1. Independent of gender, if the weather is cold and the customer less than 15 years old, he or she is likely to buy the scarf.
2. If the weather is sunny, nobody buys a scarf.
3. If the weather is warm and the person is older that 50 years, then she/he would buy a scarf.

The above rules are, of course, preliminary and will be modified or reinforced when more cases are added to the data set. When the decision tree is complete, it is possible to calculate the most frequent path and assign probabilities to the branches.

3.5.5 Bayesian networks

Probability theory is used to predict the likelihood of different outcomes, based on a set of facts, i.e., the conditional probability of an event A given another event B , which is defined as

$$P(A|B) = \frac{P(A \wedge B)}{P(B)}. \quad (3.19)$$

In general, it is difficult to obtain or estimate directly $P(A|B)$. It is simpler to calculate $P(A|B)$ by using the Bayes theorem,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \quad (3.20)$$

which predicts *a posteriori* probability of a variable, given a data set. From equation (3.20) two useful results can be extracted:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\neg A)P(\neg A)}, \quad (3.21)$$

and

$$P(A|B \wedge X) = \frac{P(B|A \wedge X)P(A \wedge X)}{P(B \wedge X)}, \quad (3.22)$$

which allows the identification of *a posteriori* probabilities using a factual data set.

Finally, the general expression for the Bayes theorem is

$$P(A = a_i|B) = \frac{P(B|A = a_i)P(A = a_i)}{\sum_{k=1}^n P(B|A = a_k)P(A = a_k)}, \quad (3.23)$$

where a_i with $i = 1, \dots, n$ is the possible set of values that could be taken by A .

Consider the network in Fig. 3.7 which shows data about customers' clothes purchases. A Bayesian network models a set of facts, under the assumption that they are related. The probability of an event can be calculated using the expression 3.20.

A Bayesian network is usually constructed by advice from human experts about

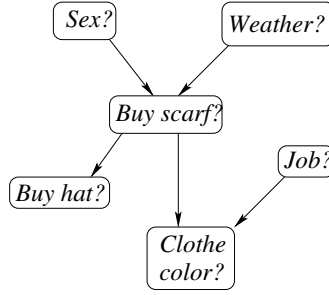


Figure 3.7: A Bayesian network for clothes purchases

a given set of facts, although the probabilities are calculated from the data. In Fig. 3.7, the expert can see that the weather and gender help define preferences. Also, the type of job can influence scarf color, whereas if the customer uses a scarf, it is possible that she or he would also use a hat.

Using the Bayesian network, rules about the purchase behavior can be identified through corresponding probabilities. For example, **if** (*sex=male and weather=rainy*) **then** *buy scarf* is true with a 80% of probability, i.e., $P(\text{buy scarf} = \text{yes} | \text{gender} = \text{male} \wedge \text{weather} = \text{rainy}) = 0.8$.

3.5.6 K-Nearest Neighbor (KNN)

Given a set of N vectors with n components, classified in m classes, a new vector is classified to the class to which the majority of the K-neighbors belong.

Let D be the set of training data, such as $D = \{(x_i, c_j) \mid x_i \in \mathbb{R}^n, c_j \in (0, 1)\}$, with $i = 1, \dots, N$, $j = 1, \dots, m$ and c_j , the class to which x_i belongs to.

In order too compare a new vector with the existing ones in D , a distance measure is required. Traditionally, the Euclidean distance is used, which is defined as

$$d_i = d(x_i, x) = \sqrt{\sum_{l=1}^n (x_{il} - x_l)^2}. \quad (3.24)$$

The basic algorithm for KNN is:

1. Given $D = \{(x_1, c_1), \dots, (x_N, c_m)\}$ and $x' = (x'_1, \dots, x'_n)$ the new example to be classified.
2. Calculate $d_i, (\forall x_i \in D)$, the distance between the new example and all the examples in the data set.
3. Let $S_k(h)$ be a function that arranges input values in descending order, returning the k highest values. Let D_x^k be the k -nearest neighbors for x' defined as $D_{x'}^k = \{(x_i, c_j) | d_i(x', x_i) \in S_k(d_i(x', x_i))\}$.
4. Assign x' to the most frequent class c_j in $D_{x'}^k$.

Fig. 3.8 shows a simple example illustrating the operation of the KNN algorithm. It shows a set of 32 training vectors ($N = 32$), with two components ($n = 2$) and classified in 3 classes ($m = 3$).

After performing steps 1 to 3 of the KNN algorithm, five examples are classified as nearest to x . Of these, three belong to γ class, one to α and the last one to β .

Using a 3-NN classifier, the winner class is β . If we use a 5-NN classifier, the winner class is γ .

The nature of the data is important; the distance should take into account that not all vectors have the same importance - indeed many contain irrelevant information.

Then a variation of the Euclidean distance is the following:

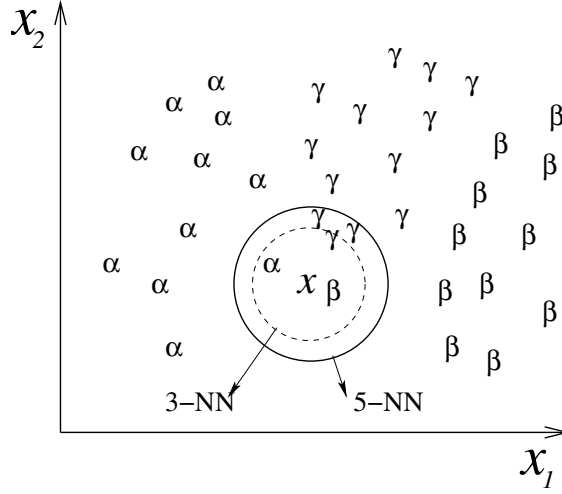


Figure 3.8: KNN classification example

$$d(x_i, x) = \sqrt{\sum_{l=1}^n w_l (x_{il} - x_l)^2}. \quad (3.25)$$

where w_l is the weight of each vector component.

3.5.7 Support vector machines (SVMs)

Support Vector Machines (SVMs) is a relatively new machine learning approach that has become very popular in many classification and regression problems in recent years due to their impressive empirical performance. SVMs were introduced by Vapnik in 1995 [222], although he was working on the same subject since the late seventies [221], but his work did not receive great attention before 1995.

The major advantage of SVMs compared to ANNs is that SVMs overcome the problem of local minima and are less prone to overfitting. “SVMs will always find a global minimum when trained” [41]. This is done by mapping the input vectors onto

a higher dimensional space from which the separation of data can be easily achieved with linear decision surfaces. So, unlike ANNs, SVMs do not have to maintain a small number of features in order to control the model complexity. SVMs scale well to high dimensional data.

The basic idea of SVMs is to map the input vectors from the low-dimensional input space on to a much higher dimensional space by using a kernel function. Quadratic programming is then used to search for the model parameters of the global optimum solution, which provides the classification model that best separates the data. The most important step in training SVMs is choosing “effective” kernel functions, by trying various kernel functions, e.g. start with a low degree polynomial kernel function and then increase the degree. This is similar to choosing the right number of hidden nodes in a neural network modelling task.

Let us consider that we are encountered with a two-class classification problem. An SVM will find the separating hyperplane that provides the highest degree of separation between the two classes. In other words, SVMs will find the hyperplane that maximizes the margin between the two classes. Fig. 3.9 presents an example in a two-feature space. If classes are not linearly separable in the given feature space, then the SVM will find the hyperplane that maximizes the margin width and, at the same time, minimizes the number of misclassified examples. In this case, some examples will fall within the margin.

From Fig. 3.9, it is easy to see that the margin width is $\frac{2}{\|\vec{w}\|}$ and we should maximize this value. The optimization problem solved by an SVM can then be formulated as follows. Given the training set (\vec{x}_i, y_i) , $i = 1, \dots, n$, where \vec{x}_i are the input vectors and the two classes being denoted with 1 and -1 (i.e., $y_i = 1$ if \vec{x}_i belongs to class 1, and $y_i = -1$ if \vec{x}_i belongs to class 2), the equation of the separating line returned by the SVM is:

$$\vec{w} \cdot \vec{x} + b = 0 \quad (3.26)$$

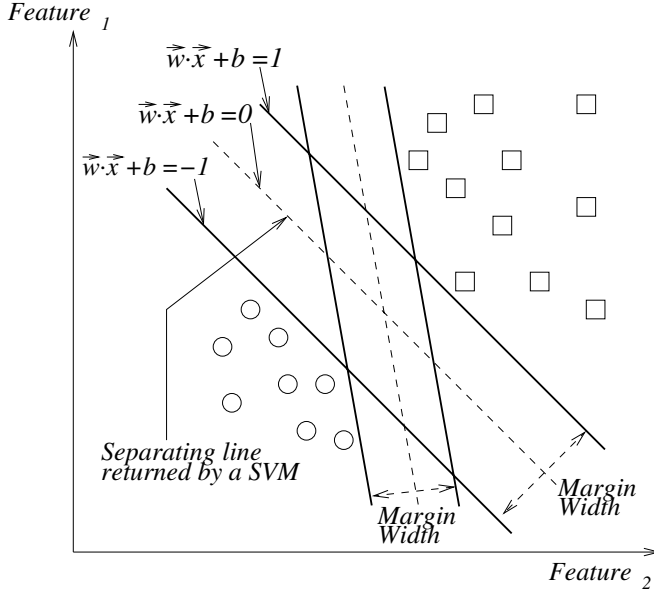


Figure 3.9: The separating surface returned by an SVM

and the SVM should find the optimal values for \vec{w}, b that provide:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\vec{w}\|^2 \\ \text{s.t.} \quad & (\vec{w} \cdot \vec{x}_i + b) \cdot y_i \geq 1, \quad \forall \vec{x}_i \end{aligned} \quad (3.27)$$

This is a convex quadratic optimization problem [28] and the SVM will be able to find the unique global minimum.

Support vector machines are based on the Structural Risk Minimization (SRM) [222] principle from statistical learning theory. Vapnik stated that the fundamental problem when developing a classification model is not concerning the number of parameters to be estimated, but more about the flexibility of the model, given by the so called “VC-dimension”, introduced by Vladimir Vapnik and Alexey Chervonenkis to measure the learning capacity of the model. So, a classification model should not be

characterized by the number of parameters, but by the flexibility or capacity of the model, which is related to the extent of how complicated the model is. The higher the VC-dimension, the more flexible a classifier is.

The SRM principle states that we should find a classification model that minimizes the sum of the training errors (called empirical risk) as well as a term that is a function of the flexibility of the classifier (called model complexity). In this way, overfitting of the model is avoided.

In conclusion, compared to ANNs and other machine learning approaches, SVMs allow to explicitly control the tradeoff between the training error and the model complexity. Due to their advantages, as described above, SVMs have become a very popular machine learning technique used in all kinds of real-world applications, including web mining, bioinformatics, process control, etc., to name only a few.

3.6 Using data mining to extract knowledge

Each of the data mining techniques reviewed above can, if properly trained using a training data set, predict behavioural situations. The prediction is made, in some cases, by applying rules to the input data (so called associate rules techniques). These rules are extracted directly from the underlying algorithms. For example, decision trees and Bayesian networks show, in a very explicit way, the rules to apply over data after the training task is completed.

A multi-layer ANN learns an approximate non-linear and multi-variable function to model a given problem, represented by a set of feature vectors. When the ANN receives a new input vector, based on what was acquired during training, it projects the new vector onto the output space (prediction).

The associations rules and multi-layer ANNs are useful tools to predict the behavior of a web user. For example, in examining the browsing behavior, the model

will predict the next web page to be visited or the session path. These techniques are used for virtual shopping; when a user is looking for a product, the web site not only identifies desired information, also displays hyperlinks for similar or associated products, preferred by other users.

Also clustering techniques are important tools for understanding human behavior. The extracted cluster patterns are not always clear or coherent, but often need expert discussion and support, particularly about how to apply the implicit rules for predictions.

From the beginning of mankind, people have been grouped in communities (e.g. clans, civilizations, countries), based on common characteristics like language, race, culture, etc. Within and between these groupings there are others divisions, for example based on individuals' socio-economic background. It would seem that the tendency to form groups with common characteristics is a common trait of mankind. This phenomenon is well known to manufacturing companies which constantly attempt to create a specific product for a particular group of consumers.

In marketing, a typical technique to discover consumer needs is the opinion poll or market sample survey. The segment sample specifies the characteristics of each product version. This process is known as "market segmentation" [137]. The segmentation is the result of clustering techniques, where the practical purpose is to find the centroid for each cluster of the product's characteristics.

The opinion survey is a technique for understanding the consumer behavior and preferences. In this sense, the web operation is like an online survey, since web logs contain registers about user preferences. Thus, the application of clustering techniques in order to understand user visits to a web site is a promising approach [62].

Users with similar behavior are grouped in the same cluster. Then, it is possible to assign a new user to one of the clusters and then predict his/her behavior based on the cluster information.

3.7 Validation of the extracted knowledge

The most important part of the validation task is to use business expertise to review the results provided by the data mining tools. Often, experts know about a given problem and perhaps even can apply a solution, but they lack the scientific foundation to hypothesize based on their expertise.

Data mining tools can help business experts to validate their hypotheses or show hidden patterns and thus demonstrate a new theory. However, it may be the case that results make little sense. If so, then the complete process should be reviewed and modified, in particular information models which are inputs to the data mining tools.

The model modifications can cause changes in the complete KDD process. Some data sources might be included and others partially or completely excluded. Changes will also occur to the information repository structure and, possibly, to all the stages involved in the data processing.

Because the KDD is an iterative process, changes can be made several times, but in each iteration a convergence should be reached. To diminish the impact of changes on the information repository, the model should be generic and allow for slow changes [113, 128]. The KDD is a never ending process because conditions change in time. For instance, the data sources could be different if the the business under study change; when we are working with variables, which change in time, for instance due to the customers purchasing behavior or simply because the market conditions have changed. However, always the extracted knowledge is potentially useful as a foundation for further work.

3.8 Mining the web

As explained in chapter 2, there is a clear difference between web data and other types of data. Data mining tools need to take into consideration the special characteristics

of data originated in the Web and perform pattern discovery tasks accordingly.

Web mining algorithms mine data which originates on the Web which they can be grouped as:

- Web content mining. The web page content is used as input by the mining algorithms. Fundamentally, the data is the free text within a web page. This process is similar to text mining and the algorithms used are similar to those used in information retrieval.
- Web structure mining. Mining algorithms use the web site hyperlink structure as input.
- Web usage mining. Web logs contain valuable information about user browsing behavior on a web site. This data can be complemented with external information like age, sex, purchasing behavior, etc.

By using web mining algorithms, significant patterns about the user behavior on the web can be extracted and, thus, improve the relationship between the web site and its users.

The main techniques and algorithms used in web mining will be explored in chapter 5.

3.9 Summary

Web data complexity was the subject of chapter 2, which also showed the need to undertake data cleaning, consolidation and preparation for the extraction of significant patterns, in order to understand user behavior on different web sites. The KDD process is a generic process and can be used for mining the web data.

The KDD process begins with the selection of data sources, which consist of web logs and the web site itself. These contain a lot of irrelevant and noisy data,

which render the reconstruction of the real web user sessions difficult. Also, there are inherent problems of processing large quantities of data from registers and, in addition, it is necessary to develop a method that automates the web data processing prior to the web user behavioral analysis.

Data prediction depends on historic information, therefore information consolidation is the next step in the KDD process. Data mart and data warehouse architectures provide a repository for past web data and will become the basis of predicting new user preferences. Commercial tools, sometimes with specific preprocessing and cleaning packages, are useful here.

Analyzing the user's browsing behavior requires special information models. Clustering methods, out of all the different data mining techniques, are the most suitable for analyzing complex data; they are the most commonly used techniques associated with the analysis of web user behavior. However, clusters should be interpreted by experts to maximize their value and who can help validate or reject the discovered clusters. This is the last step in the KDD process.

The KDD process is an iterative process which makes use of multi-dimensional information models. Given the size of the input space and large amount of webdata, information repositories are either star or snowflake model. Market conditions change over time and new knowledge is acquired, hence information needs to be continuously upgraded. Representative formats need to be loaded in the knowledge repository or Knowledge Base (KB).

Chapter 4

Web information repository

*All of the books in the world contain no more information
than is broadcast as video in a single large American city
in a single year. Not all bits have equal value.*

Carl Sagan

The Web is a primary source of data to analyze web user behavior [217]. After cleaning and preprocessing, the web data is ready to be transformed into useful information; for example, how many web users have visited the web site per month or which pages have received at least one thousand visits during the last week. This kind of information will be helpful to persons responsible for maintaining the structure and content of the web site, usually the web masters. Other potential users of this information are department managers of institutions. Many web sites show information related to one or several organization's units, for instance, a web page describing a product promotion is directly related to the marketing and sales departments. In many cases, it is easy to observe that there is a high correlation between the product publicity in the web site and the customer's answer in the traditional shop. This could open new strategies for preparing the products and services offer for both virtual and traditional customers.

While it is possible that user requirements for information met by algorithms

that work directly with web data, this often brings about difficulties, because information requirements change in time, the algorithm's maintenance is difficult, and there may be some risk involved because of programming errors made when new codes are introduced into later versions.

A **D**ecision **S**upport **S**ystem (DSS) that produces reports for predefined queries is unlikely to meet all requirements as the information user always needs varied information. So, it is necessary to develop an intermediary level of information aggregation - between primitive and final data - that provides data which will be used to answer complex information queries. This level of information is resolved with an appropriate data warehouse architecture [30, 232].

In this chapter, the theoretical foundations and practice of data warehouse architecture will be analyzed in order to create a **W**eb **I**nformation **R**epository (WIR).

4.1 A short history of data storage

From the beginning of computer science, institutions have created computer systems to store data generated by commercial transactions. Early on, data was not considered to be a strategic resource and was maintained only as storing records on for business transactions.

Hardware improvements led to important increases in data processing capabilities and so it became possible to create management reports for various commercial tasks. However, the data processing and storage capabilities continued to be problem, and reports took time to be produced.

During the 1980's, hardware capacity made a quantum jump with the appearance of multiprocessing machines with large storage capacities. This period saw the development of reports using the client/server model which allowed business users to link server computers onto the corporative **L**ocal **A**rea **N**etwork (LAN).

Until then, the data type were the traditional ones - integer, float, char, long char and, in some cases, texts. Computer capacity could provide support to operational systems¹ requirements and some OLAP (see chapter 3) applications used by the business end-users.

The next quantum jump in the computer power processing and storage abilities took place in the 1990s. Perhaps the most important advance in data storage devices was the **Redundant Array of Inexpensive Disks** (RAID)[177] system which continues to be used today. The RAID was developed by the University of California at Berkeley in 1987 and it consists of a set of hard disks organized by redundancy levels; for example, level 1 means that any data stored on a disk has a corresponding mirror in other disk. At level 0, the set of disks is fused into a unique super disk, i.e., the data is spread onto the whole set of array disks.

These key factors - the RAID technology, new computer processing capacities and new network topology for massive processing like GRID² computing [93], - now permit the storage of enormous amount of data at a relatively low cost, therefore making the advanced data processing, such as data mining, accessible to many companies.

Now, in contrast to the past, data types such as pictures, sounds, movies and, in general, any abstraction can be expressed in binary code. The Web now contains a huge amount of data in diverse forms. And, moreover, advances in data storage are unpredictable, but it can be expected that devices able to store Terabytes of data in a small space, with fast access will soon be available. The problem is how to see the wood from the trees regarding this enormous amount of data.

¹A system which supports the day-to-day business operations, for example the sales operational system in a supermarket

²A technological architecture for allowing to use a large number of computing resources on demand, no matter where they are located

4.2 Storing historical data

Many companies preserve large data banks, often containing operational data for as long as ten years. This data are used for predictions and strategic analyses. However, raw data has to be examined and used with care, some of which are specific to computer science related issues:

- Different data formats. An updating or new version of the operational program can result in a change in the data format. For example, in older sales systems, the customer information did not include cellular phone numbers whilst in new system it is included. How can old and new customer data best be combined?. Further, companies use different operational systems with different data formats, as it has been difficult, historically, to maintain a unique format for any operational system.
- Different data types. Although systems may have similar data types as internal variables, the problem is the precision by which the data is stored. For example, a system can define the type **char(8)** for storing a person's first name and another for **long char** storing the complete name. In another case, the accuracy of float numbers can generate mathematical errors that affect the final results; for example, when the price variable is stored in a **float(6,2)**, with a length of six digits and the last two floating. In another system, the same variable may be stored in **float(8,3)**. Then, when calculated, there may be errors for different levels of accuracy.
- Digitization errors. The data integrity of operational systems is checked first, e.g. if the data format accepts the corresponding float of an integer variable and whether it should be stored or rejected. More complex tests are data triggers, which are small pieces of code programmed in the operational system's language that review the contents to be stored. Finally, the user interface should have some data check points. However, mistakes such as incorrect family names,

numbers and human digitation errors, in general, are difficult to detect. If we assume that the checks were poor in the older operational system, there is likely to be unreliable information in the historical repository.

- Data conversion errors. We should consider issues such as how to combine different unit measures, for example kilograms and pounds? Again, operational systems could well have data in different units of measure.

In addition to the above points, there is the key question about which data is to be stored? There is a whole spectrum of possibilities, with some experts regarding all data as important (even if it may not be immediately useful), while there are other experts who advocate the opposite, that the only data to be stored should be the data specifically under study. Currently, there is a bias towards the first position, as we are witnessing improvements in data storage capacity and processing by the day.

Working with traditional data is a complex task and it is vital to approach data processing with considerable care. The good news is that data has a standardized structure and type; but the bad news is that with video, sound, pictures, etc., the storage process has become much more complex, as these new data types are introducing new difficulties into data storage.

All types of relevant information are contained on the Web, hence the preprocessing and cleaning task which is an important step before the web data is stored in an information repository.

4.3 Information systems

Business users, as end users, need information in order to take strategic decisions. Traditionally, information was obtained by sending requests to the company's computer department who then prepared a report and sent it to the business user. This scheme worked until the requests became more complex together with the amount of

time it took to write the extensive reports.

As most business information requests are recurrent, a logical solution is to pack queries into a system that could be operated for the business user. This helps reduce pressure on the computer department and facilitates the time required to reply to queries.

Hence corporate business information systems were born. In a short time, these have become a corporate competitive weapon [114]. With appropriate information systems, the companies can create client loyalty and solidify customer relationships. However, information systems must be agile, as both market conditions and customer change, often rendering earlier business solutions obsolete. Information systems need to be updated and often changed to newer versions. Moreover, businesses should be open to new approaches, as a strong dependence on one type of information system could be, in the long term, dangerous for the survival of a business.

Information must be stored in such a way that an information repository contains, firstly, the basic elements that allow complex queries from the business users; secondly, an interface so that business users can query the information repository and construct her/his own information system.

Much will depend on the needs for changes within the company, which in turn depends on the dynamism of the business; that is how frequently business conditions change as does the amount and quality of new data generated over time. Information dynamism on the Web is very high, so it is necessary to maintain a strict control of changes and how these changes affect business.

Although there are systems which provide various statistics about web usage [60], they are primitive and can not be used to extract significant patterns regarding virtual business; this requires the application of more advanced techniques, like web mining algorithms.

If complex queries and the web mining algorithms are needed to support business

decision making, these set the conditions for an information repository. Data warehouse architecture has been tested as a business orientated platform and its value for data mining [128, 111].

4.4 Data Mart and Data Warehouse

This architecture creates a standardized information repository [99], with characteristics which are *“subject oriented, integrated, time-variant, and nonvolatile collection of data in support of management decision making process”* [113].

While the data warehouse holds corporative information solutions, i.e., they bring together data from all operational systems in the business institution, they can be specialized and be the focus of a specific business unit as, for example, “the data mart of sales”. However the design and construction are similar and follow a common sequence [46, 55, 129, 204]:

1. Analyze end user information needs. The data warehouse must be capable of providing information for end user applications and advanced tools like data mining techniques, so that the information stored should be able to satisfy most queries. However the problem lies in what kind of questions? A simple and very useful method is by interviewing end users to understand their information needs, which in turn will define the scope of the data warehouse.
2. Data sources selection. Once the general information needs have been defined, the next step is to select appropriate data sources.
3. Develop the logical model. Two techniques are most commonly used: Cube and Star models [128].
4. Prepare a prototype for the end user. The objective is to clarify end user information requirements by showing a preliminary data warehouse model. This stage is so important in so far as the end users often do neither know their

information needs nor how to ask for it. A good remedy for this is by successive interviews with end users, demonstrating the data warehouse model, and then adjusting it to the user specifications.

5. Choose the **Data Base Manager System** (DBMS). A data warehouse can have redundancies in terms of information stored, which affect the query response time. For instance, if the end user is interested in the total daily and monthly sales of a product, both values can be pre-calculated and stored in the data warehouse to avoid time spent on the calculation. In others words, response time is the most important of the performance parameters. Then, the DBMS must meet the query response time by using the data structure to organize the data warehouse information.
6. Map the logical model in the database. This corresponds to the physical realization of the data model using data structures provided for the DBMS.
7. Store the information and evaluate the model. The **E**xtraction, **T**ransformation and **L**oading (ETL) method defines the set of steps. The first step is how to select the data sources and make the extraction. The second step shows the transformation of data into information. Finally, methods of storing information into data warehouse structures is detailed in the loading step.
8. Performance tuning. Because the query response time is a high priority, the DBMS that supports the data warehouse needs regular tuning of its internal data structures.

The data warehouse construction is iterative; the methodology is repeated several times as it is difficult to clarify the end user information needs. They should be shown prototypes, which are adjusted to deal with a broad range of information queries. At the same time the data base structures are adapted so that the stored information can be accessed easily.

Data warehouses are never completed. Sometimes it is necessary to modify structures and content incrementally - called slow changes [129] - to improve the performance or allow a new set of information issues. However by modelling the data warehouse, these modifications are possible without negative effects for the end users.

4.4.1 The multidimensional analysis

End users regard information as a series of dimensions; for example, a fact such as a sales indicator depends on markets, price, time, etc. A business question may take the form as follows: “what were the sales of pencils in the region V during the first semester 2001”. In others words, they think of the world multi-dimensionally [215]. So, particularly over the last ten years, it is business users who make the strategic decisions about purchasing software and developing software solutions oriented to business analysis and DSS [81, 100, 223].

Usually, these kinds of decisions are taken from reports developed by the information analysis area. However, to understand what kind of reports require the business users is a complex task, due to communication problems. The user needs a combination of variables in order to obtain a business indicator, but the analysis team may not understand the requirements and, even if they do, normally the user will want to make variations of the variables used to analyze, i.e., “what happen if ...” [112].

The current technology assists end users with raw file analysis. A goal in multidimensional analysis would be to give the end user an easy means of representing the information based on a common structure. The user sees a set of data repositories (tables or files) and the tools reveal a graphical interface to help create the final report. These tools can perform on line data analysis like calculations with specials formulas, or incorporate new data sets, etc.

In 1993, Codd [58] proposed the concept of *On line Analytical Proceesing* or OLAP for capturing enterprise data. Although the definition is sufficiently broad to

apply to any data format, it is more convenient to store data in files and specify how they are related.

In order to facilitate the user's work, the OLAP tools need to support the following functionalities [100]:

- **Querying.** Ability to pose powerful ad-hoc queries through a simple and declarative interface.
- **Restructuring.** Ability to restructure information in a multidimensional database, exploiting the dimensionality of data and bringing out different data perspectives.
- **Classification.** Ability to classify or group data sets in a manner appropriate for subsequent summarization.
- **Summarization/Consolidation.** This is a generation of the aggregate operators in standard SQL. In general, summarization maps multisets of values of a numeric type to a single consolidated value.

OLAP tools can be helpful to the end user, particularly where it contains information from several sources. Fig. 4.1 shows a typical business report, with sales set out along the dimensions of products and time. Here the end user is likely to want to undertake different explorations.

The data can now be used to create reports and the **Multidimensional Data Model (MDM)** [7] provides a way of storing information which can be used either for report creation or the application of pattern extraction tools.

Attributes like place, time, product are referred to dimensions while those like sales are referred to as measures. As they can be added together, the measure is called “*additive*”.

SALES			TIME												
			Year	2000				2001				2002			
			Semester	1		2		1		2		1		2	
			Month	Jan	...	Jul	...	Jan	...	Jul	...	Jan	...	Jul	...
PRODUCT	Type	City	(Cost,Sale)												
	Rice	Tokyo		(4,5)	...	(3,2)	...	(3,4)	...	(5,6)	...	(6,7)	...	(2,3)	...
		Osaka		(4,2)	...	(3,1)	...	(3,6)	...	(5,2)	...	(6,3)	...	(2,4)	...
		⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
	Fish	Tokyo													
		Osaka													
		⋮													
	Egg	Tokyo													
		Osaka													
		⋮													
	Meat	Tokyo													
		Osaka													
		⋮													

Figure 4.1: A simple sale business report

There are other cases where adding measures have no sense, e.g., when using a percentage. Here we have a “non-additive” measure. Finally, if the aggregation operation makes sense specifically when a subset of the dimensions is used, the measure is “semi-additive”.

Dimensions are usually associated with “*hierarchies*”, specifying different aggregations levels, for instance day → week → month → semester → year. In each hierarchy there is a summarization or “*aggregation point*”, i.e., a precalculated value, for example if the user asks for the monthly sales, it is not necessary to calculate a daily or weekly sum, as it has been done in a previous step, usually as a batch process. This allows for fast answers in contrast to traditional OLAP tools, which need to calculate the summarization online [223].

The MDM model must be capable of providing basic information to satisfy the end user queries at different aggregation levels. Using the above example, we may contemplate what would happen if the end user wants to know the peak sales hours of a product. As the minimum aggregation level is in days, this question cannot be

answered.

The model's granularity is the minimum amount of information to be stored in order to answer end user questions. The grain is the core of the MDM model and starting from this different aggregation levels or hierarchy is feasible. In the report shown in Fig. 4.1, a grain expression would be *"the price and cost of a product sold"*.

The MDM is developed using two techniques: cube and star. The first, as can be guessed, represents the model as a cube of information. It must be implemented in a **Multidimensional Data Base Manager System** (MDBMS), for instance Arbor Software Essbase or Oracle Express, as examples. The second technique can be developed using a **Relational Data Base Manager System** (RDBMS) like Oracle, Sybase, SQLServer, etc.

4.4.2 The Cube Model

Recalling the grain expression for Fig. 4.1, a cube representation of the sales as a function of the Time, Product and Geography attributes is shown in Fig. 4.2, as a multidimensional view. In each cell the product cost and sale price appear.

The cube is implemented in a MDBMS, which interacts through a graphical interface with the user. The creation of the hierarchy requires the definition of the aggregation point and the measure formula. Other possible operations in the cube are [100]:

Pivoting. Rotate the cube and show a particular face.

Slicing. Select one dimension of the cube.

Dicing. Select one or more dimensions of the cube.

Drill-down. Show the details of the aggregation point.

Roll-up. The inverse operation to the previous point.

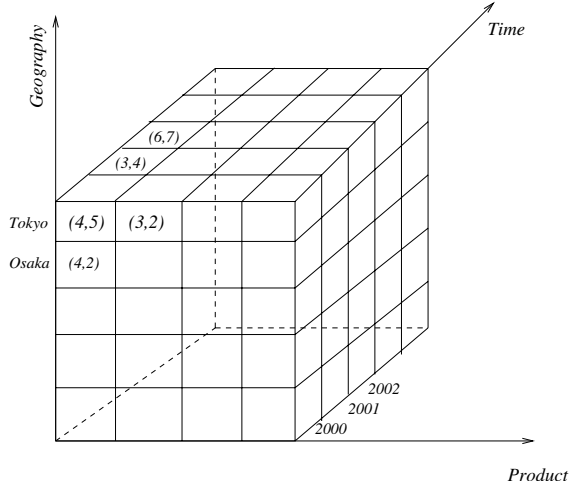


Figure 4.2: A cube model for a sale data mart

The computational cube representation uses multidimensional arrays. For instance a tri-dimensional array written as illustrated in expression 4.1

$$a : \text{array } 1 \dots \alpha_1, 1 \dots \alpha_2, 1 \dots \alpha_3 \text{ of } <TYPE> . \quad (4.1)$$

Accessing the element in position (50, 2, 1) is a simple operation like $a[50, 2, 1] \rightarrow \text{value}$.

Following the above example, a n-dimensional array is the expression 4.2.

$$a : \text{array } 1 \dots \alpha_1, \dots, 1 \dots \alpha_n \text{ of } <TYPE> , \quad (4.2)$$

with $\alpha_i \in [1, m]$, where m is the maximum number of attributes in a dimension.

Using the Fig. 4.2, if the cube model receives a request for “*what was the total cost of the product code 200 acquired in Tokyo branch for the year 2004*”, the query

would be

$$Cube[Geography.city = Tokyo, Time.year = 2004, Product.code = 200].cost \rightarrow.$$

The conceptual use of arrays has great advantages. However an MDBMS requires substantial resources, using virtual memory to implement high dimensionality. Also, other technical problems of the particular operating system version that supports the MDBMS must be considered. These problems make the high-dimensional cube implementation impractical. Anyhow, the theory underlying the model has enough mathematical validation to continue developing the MDBMS servers [7].

4.4.3 The Star Model

As the star model is implemented in a RDBMS and the **Entity Relation** (ER) nomenclature is used [57]. Formally, the ER model imposes a strict normalization with respect to **On Line Transaction Processes** (OLTPs), this arguably this's not choice for the star model, due to ER is a minimum data redundancy oriented model. However, in the star model realization, the ER is only used as a reference, making the rules behind the normalization more flexible, which creates an information oriented model. Because the RDBMS is more popular, the star model has prevailed and almost become the prototype model in the data warehouse world.

The star model consists of a single data table and a set of dimensional tables. From the ER view point, the model shows a master-detail relation. In the data table, each column represents a value. Some of them are pointers to dimensional tables (Foreign Keys) and others represent the measure value stored in the model. The dimensional tables have columns with attributes. For instance, in the product table, some columns represent the id, name and description.

Based on Fig. 4.1 and the grain expression, a star model representation is shown in Fig. 4.3.

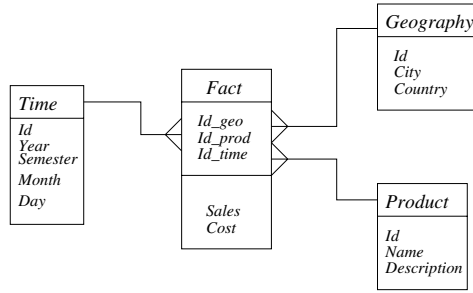


Figure 4.3: A star model for a sale data mart

The data table stores both measures present in the report: sales and cost. Using a star query that uses the time, geography and product dimensional tables, the same report shown in Fig. 4.1 can be created by executing SQL queries. For instance, in front of the request “*which was the total cost of the product code 200 acquired in Tokyo branch for the year 2004*”, the SQL query would be:

```

select cost
from fact, geography, time, product
where time.year=2004 and geography.city=Tokyo and product.code=200
/* star join */
and fact.id_geo=geography.id and fact.id_time=time.id and
fact.id_prod=product.id

```

The “star join” in the above SQL query uses the model complete set of tables. This action might consume considerable resources, reducing the performance. They increase when the dimensional tables have several attributes.

Some authors propose the normalization of dimension tables with many attributes [31, 157]. For example, we can imagine that in the dimension table Product in Fig. 4.3 the product warehouse is also stored, with its location and phone, and the product’s supplier with name and location. A normalization of this table is shown in Fig. 4.4. Following the appearance, it is called “snowflake”. This scheme is also an alternative for the data warehouse model [147].

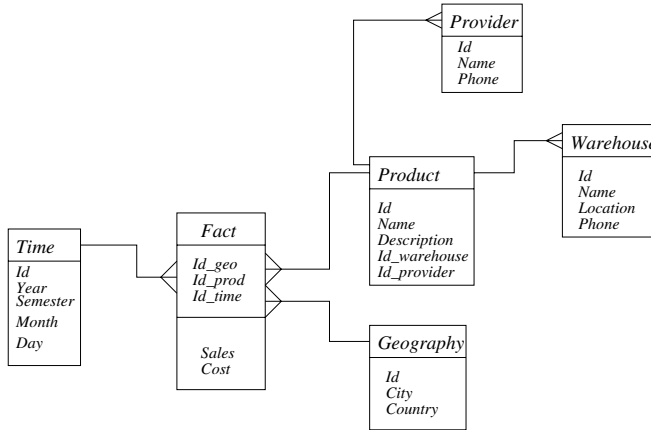


Figure 4.4: A snowflake model for a sale data mart

The star and snowflake models are the preferred models to create a data mart, using a RDBMS. In a more complex development, such as a data warehouse, the construction method involves the creation of a set data mart by always considering the final interconnection of them. This is achieved by sharing the dimensional tables among the scheme and is called a “constellation” [166].

In the last years, the commercial manufacturers of RDBMSs have increase their efforts towards improving the engine answering capabilities. A new generation of data base objects for enhanced performance has been developed, for example [82]:

Partition tables. A table can be divided physically into several partitions and each one of them stored on different discs.

Bitmap index. A data structure used to efficiently access large data sets described by low cardinality attributes.

Materialized views. A data structure that maintains in memory the answer to a query and which accelerates a reply for the next query about the same topic.

The star model has its foundation in relational algebra [100], i.e., the operations

performed have a mathematical consistency that allow the application of a formal language, like SQL, for creating the tables that will implement a practical model's realization.

4.4.4 The Extraction, Transformation and Loading Process

The ETL process groups the techniques and tools used in data extraction from multiple sources; followed by its transformation into useful information; and finally loading the data onto a repository. The ETL [99, 224] contributes to in the immense effort of constructing the data warehouse, which is a non-trivial process due to the data stemming from different origins, types, formats, etc.

Although there are software tools that automate an important part of the ETL process, human intervention is always necessary to correct any problem that has appeared. Additionally, these tools are very expensive, which is another reason for developing the ETL algorithms manually.

Usually, the complete ETL algorithms are programmed to be activated on a specific date, preferably when users of the operating system are not working, as each process uses up a lot of computer resources.

4.4.4.1 Extraction

In the extraction stage, the operational system stores the data in different architectures, such as files, relational data bases or other electronics devices and allow the creation of extraction algorithms using different platforms or languages.

An important issue is “the extraction administration”, i.e., how to prepare an extraction schedule correctly and maintain the correct algorithm version in use. Also, the data source may change, for instance a new source may be added, the format's field has updated/changed, etc. The extraction algorithms must be able to deal with

such situations.

Nor will the problem disappear as the complete warehouse environment is supported by the same architecture and will need minor maintenance and data changes throughout its life. Here, automatic tools are very useful, covering a large part of the overall process.

4.4.4.2 Transformation

The second stage of the ETL process is the most complex and time consuming task. First, the data standardization is applied; for example, if a file uses kilometers while another uses miles, one has to be modified.

A second step is to compute the measure values to be stored in the MDM. These values also consider the hierarchies, which need a summarization process of the basic information stored in the MDM. As it was introduced in section 3.3, a Data Staging Area (DSA) is defined to perform the data transformation, in which there are three basic alternatives:

1. Create a set of algorithms programmed in the native language of each data source and then merge the outputs in a unique file.
2. Use commercial tools.
3. Use the data base engine capabilities. Here the data sources are loaded onto data base structures like tables and then transformed with a proprietary language.

The first choice is expensive in programming time and prone to human mistakes, but is the default option when the data format is very specific. While the second choice has more advantages over the first, it is necessary to consider the following:

- The transformation tools are expensive.

- If the source format is too specific, the tool may not be able to read it.
- If the tool is not fully automatic, it will be necessary to program some routines by hand, which requires considerable knowledge of the tool.

The third option seems to be the most generally accepted, because most companies have a relational database engine. A common technique is to define a set of tables as DSAs. Using the engine load proprietary tools, the data sources are read and their registers inserted in the tables.

The next step is to perform the transformation process using the engine language. This process is improved by applying acceleration objects, such as indexes, buffers, etc., as well as other engine characteristics.

4.4.4.3 Loading

It might be the simplest task in the ETL process, depending on the location of the DSA and the final data format. As both are usually on the same server as the data mart or warehouse, the loading process uses the data base engine's tools and language.

A more complex situation occurs if the DSA is on an independent computer. If the environment differs from that of the data mart, then it will be necessary to define an interchange protocol, i.e., a set of steps to transmit the information from the DSA to the data mart, which also defines loading.

4.5 Web warehousing

This refers to the application of data warehouse architecture to web data [30]. The web data warehouse, or simply **webhouse**, was introduced by Kimball [128] as the solution to storing everything related to the clickstream on a web site.

The web site and the user behaviour can be distorted if changes are made to the web site's pages and structure while there is an increase in page transactions. If one takes the web site specific user behavior variability into account, it becomes necessary to develop a system that integrates the web data, past behaviour, the current site situation and techniques for making predictions.

While the data warehouse architecture is suitable for constructing the webhouse, some changes have to be made. The webhouse aims to become the information repository for analyzing the web user behavior in the Web. A similar purpose can be followed when the data warehouse is used to support operations related to the **C**ustomer **R**elationship **M**anagement (CRM). However, the data generated by the web users are quite different from the data generated by the traditional customer in a sale transaction in a traditional shop. Then the webhouse should support questions about the web user behavior, in other words, to support the **U**ser **R**elationship **M**anagement (URM), in such a way as to transform the eventual user into a new customer.

The long and distinguished history of marketing and market research has had a strong impact on the assumptions of studying customer behavior [137] has been "more data about the customer, more knowledge about his behavior" and make the effort to acquire more data at whatever the cost.

The goal is to construct a complete record about the customer, where the data warehouse plays a key role as an information repository for all customer data. The webhouse becomes the basic information repository regarding the user behavior at the web site, with the advantage, compared to traditional client information. The click stream indicates which pages have been visited by the user/customer. This is a big difference compared to the traditional customer analysis, where it is only the final purchase that is recorded and there is no information recorded regarding an interest in other products and/or services.

Although the webhouse permits an analysis of web user behavior, there is no direct personal information that allows improved browsing or preference predictions.

However, if we add specific information from operational systems, which could be contained in the data warehouse, we can construct a solid “big picture” about web site user behavior and preferences.

Some challenges in the webhouse design are:

- Web data grows exponentially in time. If the goal is to analyze all web user behavior, it is necessary to consider every users’ movements on the web site as well as changes in the site, including online modifications, which usually are implemented through dynamic web pages containing online navigation recommendations for the users.
- Response time. Due to the fact that it is very probable that the webhouse may have an automatic system as client, the response time must be under ten seconds. Similarly, if several human users request information from the webhouse at the same time from different places, then the response time must be short.

The webhouse will become an essential integrated structure, like Fig. 4.5 shows. The challenge is not minor, considering the overall response time factor. In Fig. 4.5, the web server receives user requests through the web. Depending on the request, it is possible that the requested pages may require the intervention of a **Business Application Server (BAS)**. This is the business layer in a multi-layer system.

Depending on the size of the business, it is very probable that the webhouse will reside in a big data base management system, like Oracle, DB2, Sybase, etc.

4.6 Information repository for web data

In chapter 3 we examined the needs for an architecture to maintain information from a set of data sources. The main idea is the creation of a repository where the information is stored after data preprocessing, this includes data cleaning and information

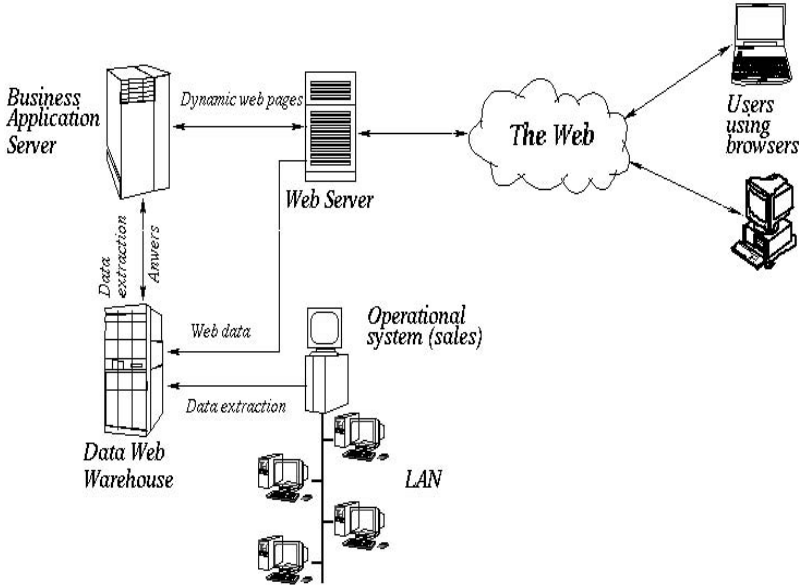


Figure 4.5: A generic data web warehouse architecture

consolidation. This repository have two objectives: to be the primary information source for end users and the input source for data mining algorithms.

Due to their complex nature, web data need special preprocessing tasks in order to clean and transform them in feature vectors stored in an information repository. An web information repository (WIR) follows the webhouse architecture, although it concentrates only on web data, specifically web logs, web text content and web hyperlinks structure.

4.6.1 Thinking the web data in several dimensions

Some examples of main types of end user questions regarding the web site use are [110]:

- Which is the most visited page in a given period?

- How much time is spent by page per session?
- How many bytes are transmitted by session?
- What is the length of the session average (in pages visited and time spent)?
- How many visits are made per page in a given period?

There may, of course, be more questions, but these examples are a starting point and they need more iterations to obtain satisfactory answers. The webhouse designer needs to respond to the fundamental question about the WIR construction, i.e., which is the grain or what constitutes the minimum information set to satisfy the business questions (either directly or by data processing)? We define the WIR grain as “*the time spent per web object visited*”. Of course, additional data can satisfy further questions and different levels of information aggregation, in order to improve the queries response time.

Because the described WIR will store data on the analysis of web site user behaviour, it is necessary to identify them either by direct ID, in the case of customers, or indirectly, in the case of visitors (anonymous web users). In the latter case, we only have approximations for visitor identification, using elements like IP, agent and access time, which are not unique in the web logs. This issue was discussed in section 2.2.1.

The second web data source is the web site represented by the content of web pages and the web hyperlinks structure. The former consists of a range of web objects (banners, emergent pages, interactive formularies, pictures, sounds, etc.) but the major interest is free text. Here, the inner hyperlinks structure (pages that point to others on the same web site) are of greater interest than external links.

4.6.2 Hyper cube model for storing web data

Based on the WIR's grain expression, the cube model structure is introduced in Fig. 4.6, which shows a generic webhouse multidimensional model to support both end user enquiries and advanced web mining tools [111, 199, 232].

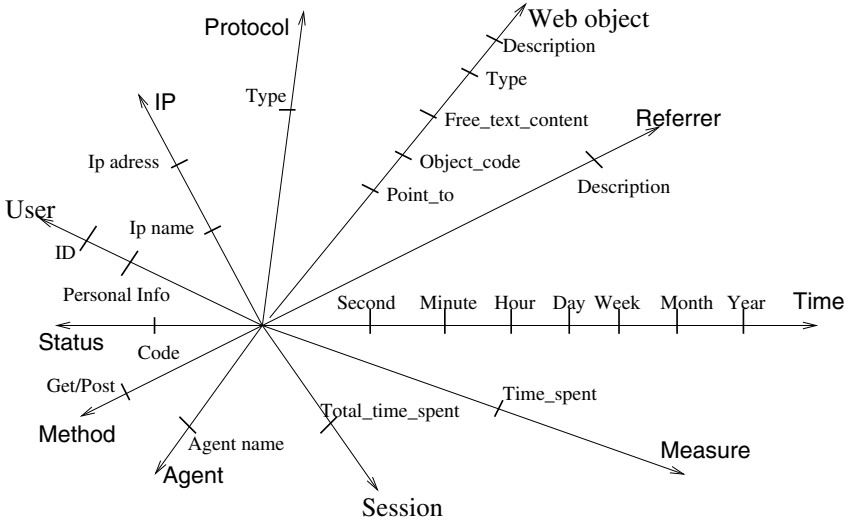


Figure 4.6: A generic cube model for web data

In Fig. 4.6, the dimensions are:

- **Web object**. Contains the web site objects and hyperlinks structure. From all web site objects in a web site, only free text are considered, for example ASP and JSP pages, Javascripts code and majority web pages written in HTML standard. These objects are stored in **object_code** and **free_text_content** attributes, respectively. The first contains the object code and the second one the free text as a set of words separated by blank spaces. Each web object contains an identification number. With this number, the hyperlinks structure can be modelled as a set of identifiers pointed to an object. The information is

stored in the **link_to** appropriate attribute.

- **Protocol**; contains the protocol used in the page transmission, e.g., http and https.
- **Status**; contains the web site object transference status.
- **Session**; contains the identified real session as a correlative number.
- **Agent**; contains the agent used to access the page, e.g., Mozilla, Opera, Explorer, etc.
- **User**; if personal information about the user is available, it would be stored here.
- **Method**; refers to the access method used to access a web page (GET and POST).
- **Time**; contains the time stamp when a web page was accessed.
- **Referrer**; in some web site log configurations, it is possible to get the object that points to the required web page.
- **IP**; contains the IP address of the page request.

Finally, the dimension contains bytes transmitted, time spent per page in a session and the amount of sessions. These measure are additive.

In the model shown in Fig. 4.6, aggregation levels and hierarchies are defined as a way to improve queries performance. For example the time dimension contains the hierarchies Minute, Hour, Day, Week, Month, and Year.

A measure is accessed in the same way as a n-dimensional array, i.e.,

WIR[*Web_object.id* = 45, *Protocol.type* = http, *status.code* = 200, *Agent.name* = mozilla,
User.id = 1002, *Session.id* = 123, *Method.type* = GET, *Time.timestamp* = 01 : 30 : 05; 05/01/04,
IP.address = 156.82.3.2].*Time_spent* →

where it is necessary to fix the coordinates of the variable, in this case “Time_spent”, in order to extract the value.

4.6.3 Star model for storing web data

After the cube model, it is easier to design a star scheme, because the dimensions are clear and it is only necessary to define the data table’s structure and its relationship to the dimensional tables. Fig. 4.7 shows the structure of a WIR using the star model.

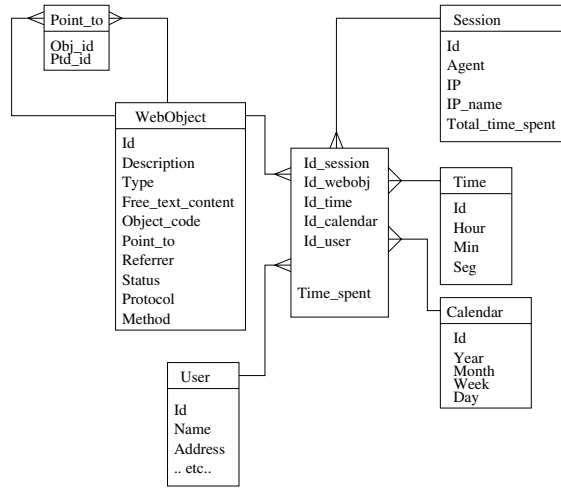


Figure 4.7: A generic star model for web data

In terms of performance, it is not convenient to define a star model with several table dimensions because the joint operation will use substantial computer resources. Also, it is important to note that the majority of the queries are unlikely to use a complete set of dimensional attributes.

The star model in Fig. 4.7 contain the following dimensional tables:

- **Web_object**; contains the web object in the web site, which is implemented with a recursive relation that provides the correct storage for the web link structure, as the final implementation will be made in a relational data base.
- **Time**; contains information relative to the time only - seconds, minutes and hour - when the web object was requested.
- **Calendar**; contains a group of hierarchies for the time when the objects were requested. This process does not require the use of the dimensional table **Time** when the question range is only about the day, week, month and year.
- **Session**; contains the user session information for identification. An important attribute is *Total_time_spent*, which stores the total time spent per session.
- **User**; contains personal information about the user.

An example of data access is the following SQL query, which extracts the average time spent per session at web object 201 during September 2004.

```
select AVG(Fact.time_spent)
from Web_Object, Session, Calendar
where Calendar.month=9 and Calendar.year=2005 and Web_Object.id=201 and
/* star join */
WebObject.id=Fact.id_webobj and Session.id=Fact.id_session
and Calendar.id=fact.id_calendar
```

Depending on the frequency with which users request the above query, it might be necessary to add a new measure with the average time spent per session in a web object.

4.6.4 Selecting a model for maintaining web data

Both cube and star models have important characteristics for WIRs. However, it is necessary to select one to implement a real solution in a DBMS. The debate about

which model should be used for the physical implementation of a data mart is centered around the kind of application implemented to perform OLAP queries [110, 223]. When the model is implemented in a RDBMS, the OLAP queries are called ROLAP for Relational OLAP; if used in a MDBMS, they are called MOLAP for Multidimensional OLAP.

The ROLAP architecture allows one to store the data in a very efficient way. Also, it is important to notice that if the institution has a RDBMS, one does not need to buy a new engine to create the data mart. The main issue of ROLAP is the data access time. In fact, it is necessary to use a large amount of resources to implement a ROLAP query with a good response time.

Given the MDBMS structure, the query to MOLAP has direct access to the data structure, allowing possible significant improvements in data retrieval and response time, when compared to a ROLAP query [61].

For the RDBMS, the response time can be improved by applying a tuning process, which defines acceleration objects, principally indexes, and efficient data storage structures, like partition tables. Also, the advanced memory structure to improve the data access time, like the materialized views, should be taken into account.

For the moment, the state of the art shows more physical developments under the ROLAP architecture, while the RDBMS has a huge support service, crucial when necessary to improve the query performance. The WIR physical structure can be implemented in a star model.

4.6.5 ETL process applied to web data

It is first necessary to identify the sources to be processed. These are the web objects with text content, web hyperlinks inner structure and web logs of a particular web site.

4.6.5.1 Processing web page text content

With web pages, extraction and transformation are executed from a data base environment. The extraction consists in selecting web objects related to free text from the web site, such as the HTML code, ASP, JavaScript, etc. The transformation process that follows is executed by a code which undertakes the following tasks:

- Assign an object identifier. Because the web site changes over time, the same file can contain new releases of an original web object, for example the home page, commonly stored in the `index.html` file. If it has changed, and the modifications not recorded on the register, it is impossible to follow the page life on site. Thus, it is important to assign a unique identifier for the web object, preferably a sequence number.
- Transform the web object text content into words. By using the vector space model introduced in section 2.3.1, the web object text content is transformed into a set of words.
- Web object description. This task may need direct human interaction. However, in some web sites that use XML, the main topic description is included.

Further, for future analysis, the tags used for each page, the object's stored code in the **web_object** dimensional table, together with the list of isolated words by object and their respective descriptions, should all be recorded and stored.

4.6.5.2 Processing the inner web site hyperlinks structure

Because the scope of the WIR is data generated at a given web site, the outer hyperlinks structure will not be considered. Thus, for each web object containing free text, a program performs searches for **href** tags and selects those which point to other free text content objects at the same site. As they have an identifier number, it is easy to

store the hyperlinks structure by using a character array, that contains the object in the first position with those pointing to internal locations. For instance, $\{1, 5, 8, 10, 2\}$ means that object “1” points to “5,8,10,2”. If the object is an edge page, the array only contains its identifier.

This process, following the codification explained above, generates a file with registers that contain the inner web site structure. The final step is loading this data onto the WIR using the Web_object dimension table, so that a recursive relation is implemented through the relational table **Point_to**, which stores the main object identifier in **obj_id** column and the pointed objects identifiers in **pointed_id** column.

4.6.5.3 Processing the web logs

Each visit to the web site leaves behind important information about the web user behavior, stored in log files [104, 123]. Depending on the traffic on a web site, these files may contain millions of records, some of them with irrelevant information, so their analysis becomes a complex task.

Web logs might contain a user’s complete web site interactions and include records for any requested object through the browser visiting the site. However, as defined, those objects that do not contain free text are regarded as excluded by the WIR. Then, the web logs has to be filtered, eliminating URL registers which do not contain references to objects with free text. This task can be implemented at the extraction stage by using a programming language, preferably one that supports regular expressions (Perl, PHP, unix scripts). Then, the web logs are filtered and the final result stored in a predefined format file.

Prior to the transformation process, in this case the web logs sessionization and the DSAs must be defined. The options are to undertake the sessionization either outside or inside of the RDBMS. The RDBMS contains data structures, acceleration objects and subroutines to manipulate huge amounts of data, so it is helpful to define a set of tables as DSA and transform data using the RDBMS proprietary language.

Following this, the web logs have to be loaded onto the DSA, using the tools of the engine. In fact, the majority of the database engines have tools for data migration between systems. The mandatory requirement is that the data sources must be created in a specific format, respecting the data types used by the RDBMS.

Fig. 4.8 shows two independent tables. The first one, called **weblogs**, receives the web logs registers. Each column has the corresponding variable to store register components.

Weblogs		Logclean	
Ip	varchar2(18)	Ip	varchar2(18)
TimeStamp	date	TimeStamp	date
Method	varchar2(20)	Bytes	number(8)
Status	number(4)	Url	number(4)
Bytes	number(8)	Agent	number(2)
Url	varchar2(20)	Session	number(4)
Agent	varchar2(20)	Timespent	number(4)

Figure 4.8: Data staging area for processing web logs

The table **Logclean** has a similar structure as **Weblog**, but the **URL** field now contains the web object identifier. The **session** field contains the session identifier and the **timespent**, stored as the time spent per object visited during the session.

The sessionization continues as described in section 2.2.1. This process could be performed by direct user identification, where there is a user identifier.

The most difficult cases are the anonymous users, here it is necessary to apply the criteria which group the logs by Agent-IP and select the registers with a time stamp that belongs to a fixed window time, by using the capabilities of any RDBMS. This is a further good reason to define the DSA as database tables.

The sessionization process is implemented as follows:

1. Select the web logs register whose url field point to web objects that contain text inside.
2. Group the web log registers by IP and Agent.
3. Sort the groups of web log registers by time stamp.
4. The session duration cannot exceed 30 minutes.
5. Because it is possible that a crawler or many users have visited the site through the same IP, it is necessary to fix the maximum number of pages visited per session. It depends on the number of pages at the site and its inner hyperlinks structure.

Finally, the loading process is undertaken by another program in the engine's language. It takes the records from **logclean** table and places them in the WIR's respective tables.

4.7 Summary

As discussed previously, the KDD process demonstrated the need to consolidate and maintain information extracted from data sources. So, an information repository seems to be the logical response, as it will allow further retrospective analysis and advanced data processing application techniques such as data mining algorithms.

Building an information repository is a non-trivial project - there are many considerations to be taken into account especially around the quality of source data. It is necessary to remember that the source data has diverse origins, part of the data can be erroneous and that any combination is likely to generate information errors.

In addition, it is necessary to consider the human point of view. The main purpose of an information repository is to satisfy end user information needs, so that its structure and content must allow for these information needs. What are end user

information requirements and how are they to be accommodated within the corporate information system? This is a complicated question, because the end user may not realize very well what they want and, moreover, new information may arise.

The star and cube models are the most valuable tools for multidimensional data analysis. Both approaches show information as measures and variables to the end user. This is a familiar approach to business, which thinks in multiple dimensions, i.e., any information report shows an indicator (measure) that depends on group of variables.

The architecture of the information repository needs to be able to consolidate and maintain the information. These characteristics are to be found in the data warehouse architecture, which is defined as a set of iterative stages. Within these stages, the prototype creation allows the end user genuine involvement in the project, assuring its success.

The data warehouse must be highly flexible to allow for slow changes when the architecture project is finished. The business can suffer unanticipated variations, so that the data warehouse design can be modified and will not need to be replaced. These characteristics make the data warehouse architecture a real information repository solution, when confronted with the complex data that originates on the Web.

A further development is the evolution of the data warehouse for web data, also called webhouse, where web site and operational system data are used as part of the data sources in the creation of the generic information repository for current enterprises, i.e., traditional and web-based companies.

By using the webhouse architecture and the star model, a Web Information Repository (WIR) was defined to store information extracted from web logs, web objects, free text and web site inner hyperlinks structure. The WIR is destined to be the main information platform for web mining algorithms and end user queries.

Chapter 5

Mining the Web

*Before a diamond shows its brilliancy and prismatic colors
it has to stand a good deal of cutting and smoothing.*

Anonymous

Web mining techniques emerged as a result of the application of data mining theory to pattern discovery from web data [52, 153, 207]. Web mining is not a trivial task, considering that the web is a huge collection of heterogeneous, unlabelled, distributed, time variant, semi-structured and high dimensional data [176]. Web mining must consider three important steps: preprocessing, pattern discovery and pattern analysis [210].

The following common terminology is used to define the different types of web data:

Content. The web page content, i.e., pictures, free text, sounds, etc.

Structure. Data that shows the internal web page structure. In general, they have HTML or XML tags, some of which contain information about hyperlink connections with other web pages.

Usage. Data that describes visitor preferences while browsing on a web site. It is

possible to find such data inside web log files.

User profile. A collection of information about a user: personal information (name, age, etc.), usage information (e.g. visited pages) and interest.

With the above definitions, web mining techniques can be grouped in three areas: Web Structure Mining (WSM), Web Content Mining (WCM) and Web Usage Mining (WUM). The following section provides short descriptions of these techniques.

5.1 Mining the structure

Web structure mining (WSM) uses the Web hyperlink structure [47, 50] to analyze hyperlinks inside web pages from a set of web sites. Web sites are considered, in this analysis, as a directed graph (see Section 2.3.2 for details and examples). Then, for example, the popularity of a web page can be examined by considering the number of other pages that point to it. So it works like a bibliography citation system, i.e., a frequently cited paper is an important paper.

Modeling the Web hyperlink structure develops the idea of a hyper-linked community or web community, which is defined as “*a set of sites that have more links (in either direction) to members of the community than to non-members*” [91]. This information can be used to obtain the relative importance of a page in the web community and answers the question on “how popular is our web site on the Web?”.

Search engines, like **Google** or **Yahoo!**, define importance by the number of times a page is cited for a particular subject by using algorithms such as **HITS**¹ and **PageRank** [136]. Both algorithms rank the pages by identifying its relative weight in the web community.

Some assumptions used in classifying web pages are:

¹Hypertext Induced Topic Selection

- A credible page will point to credible pages.
- The hyperlink names have some meaning.
- The page ranking depends on the user query and the hyperlink structure.

A basic model for a web community is the graph $G(P, L)$, where $P = \{p_1, \dots, p_n\}$ is the set of pages and $L = \{l_1, \dots, l_m\}$ is the set of hyperlinks that interrelate the pages (see Section 2.3.2). Then the real importance of a page within the community can be decided, i.e., if the page is authoritative, a hub or irrelevant. The last two descriptors do not contain important information for the whole community as they refer to pages that are only pointed to by others on the same web site.

5.1.1 The HITS algorithm

indexAlgorithm!HITS Given a query, this algorithm finds pages with text content that represent a relevant and good information source, and ranks them according to their importance in the web community [130].

To calculate page relevance, pages are grouped as either authoritative and hub. Authoritative pages are natural information repositories, i.e., pages whose content is important in the community. Hub pages concentrate a set of hyperlinks to other pages.

The HITS algorithm assumes that the authority comes from in-edges. A good authority comes from good hubs whereas a good hub contains links that point to good authorities. A simple method to differentiate page relevance is to first assign non-negative weights, depending on whether a page is hub or authoritative. Then, preliminary weights are adjusted by an iterative process and the page-relative importance in the community can be calculated [131].

Let a_p and h_p be the weights associate to authoritative and hub pages, with $p \in P$. These weights can be calculated as

$$a_p = \sum_{\forall q, p \in P/q \rightarrow p} h_q, \quad (5.1)$$

$$h_p = \sum_{\forall q, p \in P/q \rightarrow p} a_q, \quad (5.2)$$

where $q \rightarrow p$ indicates that there exists a hyperlink from q to p . These equations define an iterative algorithm; the variable initialization assigns a non-negative weight to a_p and h_p for each page p , for example the value 1. Then, the weights are updated by repeating equations 5.1 and 5.2. When the algorithm stops, the authoritative pages are returned by number order of hub links.

Let $A = (a_1, \dots, a_n)$ and $H = (h_1, \dots, h_n)$ be the vector containing the weights for authority and hub pages in the community respectively. The expressions 5.1 and 5.2 can be rewritten using a matrix representation as $A = M^T H$ and $H = M A$, where

$$M = (m_{ij}) = \begin{cases} 1 & \text{if page } i \text{ points to } j \\ 0 & \text{otherwise} \end{cases} \quad (5.3)$$

The iterative formulas to calculate A and H are:

$$A^{(k+1)} \leftarrow M^T H^{(k)} = (M^T M) A^{(k)}, \quad (5.4)$$

$$H^{(k+1)} \leftarrow M A^{(k)} = (M M^T) H^{(k)}, \quad (5.5)$$

where $A^{(k)}$ and $H^{(k)}$ are the matrices at the k -iteration of the algorithm. The HITS algorithms is given below:

1. Initialize $A^{(0)} = (1, \dots, 1)$ and $H^{(0)} = (1, \dots, 1)$.
2. Calculate $A^{(k+1)} = M^T M A^{(k)}$ and $H^{(k+1)} = M M^T H^{(k)}$.

3. Normalize $A^{(k+1)}$ and $H^{(k+1)}$.
4. If $\|A^{(k+1)} - A^{(k)}\| < \delta_1$ and $\|H^{(k+1)} - H^{(k)}\| < \delta_2$ stop.
5. $H^{(k+1)} = H^{(k)}$ and $A^{(k+1)} = A^{(k)}$. Go to point 2.

As an example of HITS's operation, consider Figure 5.1, which shows a simple web graph with four nodes ($n = 4$).

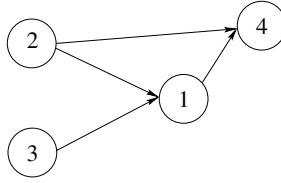


Figure 5.1: A simple web-graph for a web community

In this case, the matrices M and M^T are:

$$M = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \text{ and } M^T = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix},$$

then

$$M^T M = \begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 \end{pmatrix} \text{ and } M M^T = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

After initializing $A^{(0)} = H^{(0)} = (1, 1, 1, 1)$, we have the following sequence of normalized values:

$$A^{(1)} = \begin{pmatrix} 0.5 \\ 0 \\ 0 \\ 0.5 \end{pmatrix}, A^{(2)} = \begin{pmatrix} 0.4 \\ 0 \\ 0 \\ 0.6 \end{pmatrix}, A^{(3)} = \begin{pmatrix} 0.47 \\ 0 \\ 0 \\ 0.53 \end{pmatrix}, A^{(4)} = \begin{pmatrix} 0.49 \\ 0 \\ 0 \\ 0.51 \end{pmatrix}$$

and

$$H^{(1)} = \begin{pmatrix} 0.33 \\ 0.44 \\ 0.22 \\ 0 \end{pmatrix}, H^{(2)} = \begin{pmatrix} 0.37 \\ 0.43 \\ 0.2 \\ 0 \end{pmatrix}, H^{(3)} = \begin{pmatrix} 0.37 \\ 0.43 \\ 0.19 \\ 0 \end{pmatrix}$$

The HITS algorithm focuses on a subgraph of the web, because it works with a set of pages the content of which is related with the query. Then the pages' weights are calculated by successive approximations. In general, the algorithm will converge after few (around five [49]) iterations.

The HITS algorithm, after using the query to extract the pages, ignores the text content when ranking pages. The algorithm is purely a hyperlink-based computation.

In this sense, the CLEVER system [48] addresses the problem by considering the query's terms in the calculations implied by equations 5.1 and 5.2, and assigns a non-negative weight to each link with an initial value based on the text that surrounds the hyperlink expression (href tag in HTML) (more details in [50]).

5.1.2 The Page Rank algorithm

The Page Rank algorithm extracts the relevant pages from the web graph independent of the query [175]. The algorithm is like a random walk surfer, who starts at a random page in the web-graph and then has a choice - to follow a forward link or to browse an unrelated page.

By construction, the algorithm does not depend on the query for ranking pages, i.e., the authorities and hub pages weights are calculated off-line, independent of the future user queries.

The assumption is that the importance of a page is given by the number of pages that point to it.

To calculate the importance (x_p) of a page “p” in the web graph, the equation 5.6 is a basic expression of Page Rank algorithm on how to combine the importance (x_q) of the page “q” that points to “p”.

$$x_p^{(k+1)} = (1 - d) \frac{1}{n} + d \sum_{\forall q, p \in P / q \rightarrow p} \frac{x_q^{(k)}}{N_q} \quad (5.6)$$

where n is the number of pages in the web graph, N_q is the amount of out-links of q , d is the probability that the surfer follows some out-links of q after visiting that page and $(1 - d)$ the probability of visiting other pages that do not belong to the q 's out-links.

The Page Rank algorithm initializes $x_p^{(0)} = 0, \forall p \in P$, with n the number of nodes in the whole web-graph.

Equation 5.6 can be rewritten using a matrix representation, as follows:

$$X^{(k+1)} = (1 - d)D + dMX^{(k)} \quad (5.7)$$

where $X^{(k)} = (x_{p_1}^{(k)}, \dots, x_{p_n}^{(k)})$, $D = (\frac{1}{n}, \dots, \frac{1}{n})$ and $M = (m_{ij} = \frac{1}{N_j})$, with N_j the amount of out-links from page j , such as $j \mapsto i$.

The Page Rank algorithm is given below:

1. Initialize $X^{(0)} = (0, \dots, 0)$ and $D = (\frac{1}{n}, \dots, \frac{1}{n})$.
2. Set the d parameter (usually $d = 0.85$).
3. Calculate $X^{(k+1)} = (1 - d)D + dMX^{(k)}$.
4. If $\|X^{(k+1)} - X^{(k)}\| < \epsilon$ stop and return $X^{(k+1)}$.

5. Else $X^{(k+1)} = X^{(k)}$ and go to point 3.

To show PageRank's operation, consider again the Fig. ?? . Then

$$M = \begin{pmatrix} 0 & 0.5 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0.5 & 0 & 0 \end{pmatrix} \text{ and } D = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}.$$

Initializing $X^{(0)} = (0, 0, 0, 0)$, we have the following sequence:

$$X^{(1)} = \begin{pmatrix} 0.0375 \\ 0.0375 \\ 0.0375 \\ 0.0853 \end{pmatrix}, X^{(2)} = \begin{pmatrix} 0.0853 \\ 0.0375 \\ 0.0375 \\ 0.126 \end{pmatrix}, X^{(3)} = \begin{pmatrix} 0.0853 \\ 0.0375 \\ 0.0375 \\ 0.126 \end{pmatrix}.$$

The PageRank algorithm's main advantage is its independence from the terms of the query, and search engines like Google use it for their page ranking processing. Google probably uses a variation of the basic PageRank algorithm, possibly related to the number of matching query terms, location of matching, etc. The exact expression has not been made public.

PageRank's disadvantages concern its assumptions:

- If a page points to another page, it receives a vote in the PageRank calculus.
- If a page is pointed to by many pages, it means that the page is important.

This assumes that *"only the good pages are pointed to by other good pages"*. However there may be several reasons as to why this is the case, for example;

- Reciprocal links. If page A links page B, then page B will link page A.
- Link requirements. Some web pages provide "electronic gifts", like a program, document, etc., if another page points to it.

- Communities based on close personal relationships. For example, the pages of friends and relatives where their pages indicate the pages of other friends or relatives, because of human relationship between them.

There are probably many other cases, where the PageRank calculations could be contaminated by situations that do not reflect the real importance of a web page in the community.

5.1.3 Identifying web communities

The explosive growth of the Web has many positive benefits for the spread of the human knowledge, not least that it allows global access to the information contained in web pages. However the content analysis of the pages has proved to be difficult due to the decentralized and unorganized nature of the Web. Hence the idea of the Web community to reduce the difficulties of the information search task [90].

The web community identification has several practical applications, for it allows targeted search engines, content filters, and complement of text-based searches [211]. However the most important is the analysis of the entire Web for studying the relationship within and between communities - science, research and, *Social Networks* in general, [141, 160].

If the Web is represented as a directed graph or web-graph, then web communities can be identified by the “Max Flow-Min Cut” method [92], [91]. Let $G = (V, E)$ be the directed web-graph that represent the Web. A web community is defined as a set of vertexes $U \subset V / \forall u_i \in U, \exists u_j \in U / u_i \rightarrow u_j$ with $i \neq j$ and $u_j \notin (V - U)$.

The Max Flow-Min Cut theorem resolves the $s-t$ maximum flow problem defined as follow. Let $G(V, E)$ be a directed graph with the edge capacities $c(u_i, u_j) \in \mathbb{Z}^+$ and two vertices $s, t \in V$; find the maximum flow that is able to route from s to the sink t , maintaining capacity constraints.

In [91], it is demonstrated that a web community can be found by calculating the $s - t$ minimum cut of G , s and t being the source and the sink vertices in G , respectively.

Figure 5.2 shows the web community identification process.

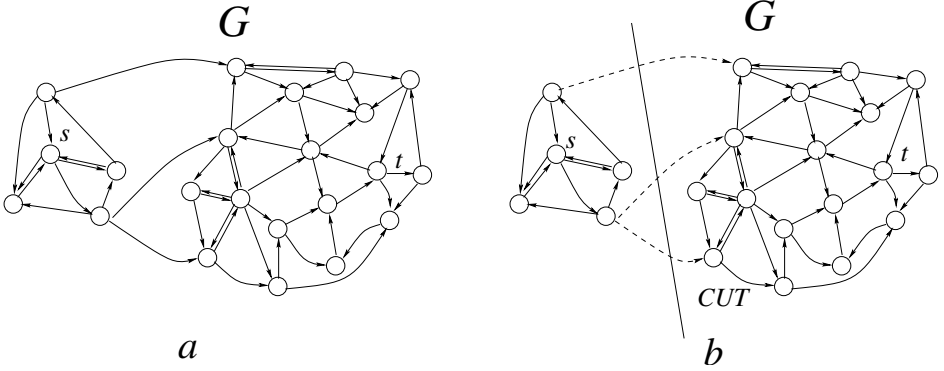


Figure 5.2: A web community identification by using Maximum Flow method

The maximum flow method will split the entire graph G in two subgraphs, using a vertex s in the left side and a vertex t in the right side of G (see figure 5.2(a)). Then, the three dashed links are removed (see figure 5.2(b)).

5.2 Mining the content

The objective of mining the web content is to find useful information from web documents. In this sense, it is similar to Information Retrieval (IR) [14] techniques. However, web content is not limited to text, but includes other objects like pictures, sounds and movies. Hence, it is possible to consider IR as a particular case of Web Content Mining (WCM).

There are two main strategies in WCM: the mining of document contents (web

page content mining) and improving the content search with tools such as search engines (search result mining).

A web text analysis can be unstructured as free text, semi-structured as HTML or fully structured as a table or database [176]. Using these types of data, the knowledge extraction can be represented to understand user preferences [154]. For instance, understanding which words or concepts are more important to the user may help to create new contents for web pages. As well as analyzing words contained in web pages, the technique extracts concepts from the hyperlinks [47].

Before applying any WCM technique, it is necessary to transform the web page into a feature vector. This can be done by using the vector space model introduced in section 2.3.1, with variations for the special words inside a web page, for instance the marked words between two html tags. For WCM purposes, let $P = \{p_1, \dots, p_Q\}$ be the set of Q pages considered in the mining set. A page p_i is represented by a vector $wp^i = (wp_1^i, \dots, wp_R^i) \in WP$, where WP is the set of feature vectors for P and R is the number of words in the entire set of pages after a cleaning stop words and stemming process. The next sections review the most important WCM techniques. The algorithms analyzed use web documents as an input to a vectorial representation.

5.2.1 Classification of web page text content

A text classifier is a function that receive a document “ d ” as input and returns a scalar value with a category $c_i \in C$, such as $\bigcup c_i = C$ [203]. The function is known as “Categorization Status Value” $CSV_i : D \rightarrow [0, 1]$, i.e., given a document d , return a value between $[0, 1]$, which classify d in the category c_i . The $CSV_i(d)$ takes different expressions, according to the classifier in use; for example, a probability approach [149] based on a Naive Bayes theorem or a distance between vectors in a r -dimensional space [201].

An important aspect of the classification is the minimum threshold τ_i for d to

be classified under c_i category. If $CSV_i(d) \geq \tau_i$ then $d \in c_i$ category, or it will be necessary to continue reviewing other categories.

In general the text classification process can be seen as a category search task [115], i.e., given a training set of documents with known characteristics, the algorithm is looking for one or more categories to which the document belongs.

In the early years text classification was undertaken by hand, but quickly replaced by semi-automatic or full-automatic [13] approaches, like Nearest Neighbor [143] Bayesian models [158], Support Vector Machines [120], Artificial Neural Networks [109, 193] and Decision Trees [11].

The above mentioned algorithms can be used in the classification of web pages, but with variations. For instance, web page tags contain valuable information about the page content which can be used in the categorization process. Web page classification algorithms can be grouped as [13]:

- Manual categorization: The simplest way is to classify the web pages manually, by using domain experts. Classification is subjective and the method cannot be applied with a large number of documents.
- Applying clustering approaches. Previous to classifying the web pages, a clustering algorithm is used to find the possible clusters in the training set. A domain expert rejects or accepts the clusters and proposes various categories.
- Meta tags use the information contained in the web page tags (`<META name="keywords">` and `<META name="description">`).
- Text content based categorization. The text content is transformed into feature vectors, often by using a variation of the vector space model. Then the algorithm uses the feature vectors for classification.
- Link and content analysis are based on the fact that the hyperlinks contain the information about which kind of pages are linked (href tag).

5.2.2 Clustering for groups having similar web page text content

Efficient search tasks and semi-automatic or fully-automatic document categorizations can be undertaken when pages are grouped.

Clustering techniques need a similarity measure to compare two vectors by common characteristics [213]. In the case of web pages, using the vector space model, one page $p_i \rightarrow wp^i = (wp_1^i, \dots, wp_R^i)$ can be compared with another by using a simple distance like the angle's cosine between two vectors, i.e.,

$$distance(wp^i, wp^j) = \frac{\sum_{k=1}^R wp_k^i wp_k^j}{\sqrt{\sum_{k=1}^R (wp_k^i)^2} \sqrt{\sum_{k=1}^R (wp_k^j)^2}}. \quad (5.8)$$

More complex and semantic based similarity measures have been proposed [213]. Let $C = \{c_1 \dots, c_l\}$ be the set of “ l ” clusters extracted from WP . In the “hard” clustering $\exists! c_k \in C, /wp^i \in c_k$; in “soft” clustering, a vector wp^i can belong to more than one cluster [125, 138]. Several document clustering algorithms have been proposed in the last years [87, 248], mainly as a means of improving the performance of the search engine by offline grouping of document contents [119]. An interesting approach is the utilization of K-means and its variations in overlapping clusters, known as Fuzzy C-means [117]. In the WEBSOM project [109], an artificial neural network is used to group documents extracted from Usenet-newsgroups, while in the OPOSUM project [212] a balanced clustering is implemented by using multi-level and multi-constraint graph partitioning. Statistical methods have also been applied, for example, mainly based on Naïve Bayes theorem. Practical applications are key-phrase extraction [220] and hierarchical clustering [108].

5.2.3 Some applications

The WCM techniques are used mainly to improve web document search by categorization and identification of the topics represented in a web document. Applications of WCM will be analyzed below.

5.2.3.1 WEBSOM

WEBSOM is “*an explorative full-text information retrieval method and browsing tool*” [109] based on Self Organizing Maps. Some application demonstrations are shown at <http://websom.hut.fi/websom/>.

WEBSOM organizes free text documents by the application of an automatic procedure. Before using WEBSOM, the documents must be transformed into feature vectors. Here it is very important to apply an exhaustive stop words cleaning and stemming process, as high vector dimensionality requires a lot of computer resources which affects the convergence of the SOM [144].

The WEBSOM works as follows. The entire set of documents are mapped onto SOM units. Documents that are similar, based on their content, will belong to neighboring SOM map units. From each map unit, document pointers are defined. Thus, when a search task is performed, the documents that match the search expression best are extracted by identifying the SOM unit and then by the selection in the group of documents indicated by the unit.

In order to facilitate the examination of a topic, WEBSOM implements an interface whereby the user can explore the map. The map units are labelled and when the user selects one, it displays information related to the topic.

5.2.3.2 Automatic web page text summarization

The goal is to automatically construct summaries of a natural-language document [103]. In this case, a relative semi-structure is created by the application of HTML tags from web page text content, which examines topics without restriction to a domain. In many cases, the pages might only contain few words and non-textual elements (e.g. video, pictures, audio, etc.) [9].

In text summarization research, the three major approaches are [156]: paragraph based, sentence based and using natural language cues in the text.

The first approach consists in selecting a single paragraph of a text segment [161] that addresses a single topic in the document, under the assumption that there are several topics in the text. The application of this technique in a web page is not obvious; web site designers have a tendency to structure the text by paragraph per page. Therefore a document contains only a single topic, which makes the application of this technique difficult.

In the second approach, the most interesting phrases or key-phrases are extracted and assembled in a single text [56, 255]. It is clear that the resulting text may not be cohesive, but the technique's goal is to provide maximum expression of the information in the document. This technique is suitable for web pages, since the input may consist of small pieces of text [43]. The final approach is a discourse model based on extraction and summarization [145, 152] by using natural language cues such as proper names identification, synonyms, key-phrases, etc. This method assembles sentences by creating a collage text consisting of information about the entire document. This technique is most appropriate for documents with a specific domain and thus its implementation for web pages is difficult.

5.2.3.3 Extraction of key-text components from web pages

The key-text components are parts of an entire document, for instance a paragraph, phrase and a word, that contain significant information about a particular topic, from the web site user point of view. The identification of these components can be useful for improving the web site text content. In this context, a web site keyword is defined as *“a word or possibly a set of words that make a web page more attractive for an eventual user during his/her visit to the web site”* [228]. The same keywords may be used by the user in a search engine, when he or she is looking for web content.

In order to find the web site keywords, it is necessary to select the web pages with text content that is significant for users. The assumption is that there exists a relation between the time spent by the user in a page and his/her interest in its content [242]. Usually, the keywords in a web site are correlated with “most frequently used words”. In [43], a method for extracting keywords from a large set of web pages is introduced. The technique is based on assigning importance to words, depending on their frequency, in all documents. Next, the paragraph or phrases that contain the keywords are extracted and their importance is validated through tests with human users.

Another method, in [17], collects keywords from a search engine. This shows the global word preferences of a web community, but no details about a particular web site.

Finally, instead of analyzing words, in [154] a technique to extract concepts from web page texts are developed. The concepts describe real-world objects, events, thoughts, opinions and ideas in a simple structure, as descriptor terms. Then, by using the vector space model, the concepts are transformed into feature vectors, allowing the application of clustering or classification algorithms to web pages and therefore extract concepts.

5.3 Mining the usage data

The interest in Web Usage Mining (WUM) is growing rapidly in the scientific and commercial communities, possibly due to its direct application to web personalization and the increased complexity of web sites [22, 64]. In general, WUM uses traditional data mining methods to deal with usage data. However, some modifications are necessary due to the different types of usage data.

The first step is preprocessing due to usage data are contained in web logs, characterized by a great deal of incompleteness [210]. The possible scenarios are:

- Single IP address/Multiple Server Sessions. Usually, an Internet Service Provider (ISP) provides mechanisms to accelerate the visitor request to a web site, for instance Proxy Servers. In this case, a single IP address is accessing the web site, but there may be several real sessions per request.
- Multiple IP address/Single Server Sessions. For privacy or ISP configuration, it is possible to give a random IP address to a visitor request.
- Multiple IP address/Single Visitor. A visitor that accesses a web site from different machines but always has the same behavior.
- Multiple Agent/Single User. As in the last case, when a visitor uses different machines, but uses different agents.

By applying the sessionization process (see section 2.2.1), it is possible to correct, at least in part, the problems detected and find the real session for each visitor. This assumes that the preprocessing step returns the basic input data for the mining process.

The goal of WUM is to discover patterns using different kinds of data mining techniques, (statistical analysis, association rules, clustering, classification, sequential patterns and dependency modelling [123, 136, 210]).

Each WUM technique needs a model of user behavior per web site in order to define a feature vector to extract behavior patterns. Usually, the model contains the sequence of pages visited during the user session and some usage statistics, like the time spent per session, page, etc. WUM approaches for usage extraction patterns are reviewed in the following section.

5.3.1 Statistical methods

Several tools use conventional statistical methods to analyze user behavior in a web site². All use graphics interfaces, like HITSograms, containing statistics about the web site's usage, for example, the number of clicks per page during the last month.

By applying conventional statistics to web logs, different kinds of analysis [37] can be achieved, e.g., which is the most frequent item accessed, pages never visited, time spent for the users during the session, etc. These basic statistics are used to improve the web site structure and content in many commercial institutions [73]. Because their application is simple, there are many web traffic analysis tools available and it easy to understand the reports produced by these tools. Usually a tool is configured to generate an automatic periodic report, which can contain limited low-level error analysis, for example, the detection of unauthorized access or alerts about pages that do not exist in the web site, but are requested by users.

Simple statistical analysis solves many problems associated with web site performance, by improving the access security and, in general, supporting simple web site modification decisions.

5.3.2 Clustering the user sessions

Clustering user sessions can be used, essentially, for grouping users with common browsing behavior and those interested in pages with similar content [210]. In both

²<http://www.accrue.com/>, <http://www.netgenesis.com>, <http://www.webtrends.com>

cases, the application of clustering techniques is straightforward when extracting patterns to improve the web site structure and content. Normally, these improvements are carried out at site shutdown. However, there are systems that personalize user navigation with online recommendations about which page or content should be visited [181, 238].

The most important feature of a clustering technique is the similarity measure or method used to compare the input vectors for creating the clusters. In the WUM case, the similarities are based on usage data at the web site, i.e., the set of pages used or visited during the user session. Because there are several usage vector representations, each similarity measure must consider the particular user navigation model expressed in the feature vector.

In [252], the usage vector is represented by the set of pages visited during the session; the similarity measure considers the set of common pages visited during the user session. Following this reasoning, in [123], the pages visited are also considered for similarity with reference to the structure of the web site as well as the URLs involved.

In [106] and [195], the sequence of visited pages is incorporated in similarity measure additional to the page usage data. In [233], it is proposed that, besides the page sequence, the time spent per page during the session and the text content in each page are included in the similarity measure.

Several clustering algorithms have been applied to extract patterns from web site usage data [138]. In [235], a Self Organizing Feature Map with a toroidal topology is used to analyze user behavior. The toroidal topology maintains the continuity of the navigation, and allows the extraction of more significant clusters than in the open topology case.

Another interesting approach is the ant clustering algorithm proposed in [2], where the ant colony behavior and their self-organizing capabilities are used to model

the user browsing behavior on a web site.

A conceptual clustering approach [88, 89] is found in [179] for tackling the index page synthesis problem. A page index is a page whose content is a set of links to other pages covering a particular topic. Conceptual clustering allows the development of content links which can suggest changes to the web site.

In [123, 195], the C-means clustering method [18] is used to create navigation clusters. This approach generates overlapping clusters, where a user session could belong to several clusters with different “*grades of belonging*”. This approach is useful for usage pattern extraction, allowing a user to belong to different clusters based on his or her interests.

5.3.3 Classification of the user behavior in a web site

Before performing a classification task, it is necessary to define the predetermined classes to map the input data. In web usage mining, the classes have a direct correspondence to different user profiles. Classification is defined by using selected features that describes each user’s behaviour [83].

Decision rule induction is one of the classification approaches widely used in web usage mining. In [171] HCV, a heuristic attribute-based induction algorithm is used for classifying the pages visited and the keywords used by the user for searching tasks. The practical result is a set of rules that represent the users’ interests.

In the decision tree induction approach, the pages visited by the user are used as Page Interest Estimators (PIE). Here a tree is created by recursive partition of a URL visited by the user and registered in the web log files, where the branches from the root to the leaves define a membership relationship for a particular navigation class. The trees can be constructing by using several algorithms, such as C4.5 [186]. Another interesting approach is the Naïve Bayesian classifier. It uses PIEs to predict if a page is interesting for a certain user, based on his previous navigation on the site

[51].

Fuzzy Neural Networks (FNNs), in [254], is a classification technique which offers personalized web-based systems for web users. FNN defines classes in the web site hyperlink structure and classifies the users according to their navigation behavior.

5.3.4 Using association rules for discovering navigation patterns

Association rules represent relations between elements in a database, that occur during a determined event, (e.g. a user clicked a web object inside a web page). The common model used for representing the association rule is $A \Rightarrow B$ (more details in Section 3.4.2.1).

In WUM, the association rules are focused mainly on the discovery of relations between pages visited by the users [163]. For instance, an association rule for a Master and Business Administration (MBA) program is

mba/seminar.html \Rightarrow *mba/speakers.html*,

showing that a user who is interested in a seminar tends to visit the speaker information page. Based on the extraction rules, it is possible to personalize web site information for a particular user [164].

From a different point of view, in [202], a Bayesian network is used to define taxonomic relations between topics shown in a web site. A Bayesian network is an acyclic graph that represents the joint probability distribution. In this case, each node represents a stochastic variable related to a topic where the probability is calculated by the level of interest a user has in the topic. The network's arcs represent the topics dependence probability. A Bayesian network, using the data content in the web logs, is created and it is possible to extract association rules for user navigation behavior on a web site from its structure.

Another approach uses fuzzy association rules for web access path prediction

[250]. The method applies to the case-based reasoning approach on user sessions extracted from web log files. Here, time duration is include as an attribute of the web access.

5.3.5 Using sequence patterns for discovering common access paths

Sequence patterns are used to discover frequent subsequences in a set of sequential data, to identify frequently occurring temporal patterns. In the case of WUM, the main idea is to find sequential navigation patterns in user sessions, for example that 60% of the users who visited *mba/index.html* and *mba/speakers.html*, visited *mba/seminar.html* in the same session.

Two methods have been used to analyze sequential patterns; deterministic and stochastic techniques. The former approach uses the association rules technique, and indeed some association rules algorithms have been adapted to extract sequential patterns [167]. Here user navigation behavior is used for sequential patterns discovery, for example in the case of the Web Utilization Mining tool[208], where sequential rules are extracted from web logs. Following the same line of research, in [54], the *Maximal Forward References* concept uses *the sequence of documents requested by the user up to the last document before backtracking*. It aims to transform the original sequence into a set of transversal patterns.

Another approach for sequential pattern extraction uses clustering techniques. In [243], a clustering algorithm is developed to extract significant user navigation behavior patterns by the identification of the centroid in each cluster. Then, sequential rules can be created to implement online navigation recommendations to the user.

The common stochastic approach for sequential pattern discovery applies Markov Models. In [29], the model is used to predict a page that a user might visit during his session. In [259], a probabilistic distribution model is applied to the user's visit

and subsequent referrer pages. The first is defined as the current page visited by the user and the second as the previous page which the user accesses to reach the current page. With this information, the model predicts the next page to be visited by the user. Another predictive approach for subsequent visits uses hypertext probabilistic grammar [35], where the web pages are expressed with “*non-terminal symbols*”, the links between them by “*production rules*” and the page sequence by “*strings*”. Using this grammar, an algorithm can be applied in order to identify the user’s navigation patterns.

5.3.6 Some particular implementations

Because the WUM techniques are suitable for analyzing user navigation behavior in a web site, there are many ways to apply them in real-world cases. Not only have academic institutions applied these techniques to web mining research, but commercial companies have seen the potential of WUM for adapting the structure and content of their web-based systems, and so improve their relationship with the web users. The next subsections will examine some important practical initiatives regarding the utilization of web mining techniques.

5.3.6.1 Web query mining

One method for helping the web visitor look for information is to use a local search engine for indexing the web site. This system acts as a simple interface, often a tool box which receives the user’s queries, usually a set of keywords, and returns a page with the hyperlinks indicating the pages that contain the requested keywords. The method can be used iteratively by successive queries that refine the information needs by keywords. In each query, we can find valuable information about what the users are looking for, because the keywords used are related with the particular user’s information needs.

Usually the query's content, i.e., the keywords used, is normally stored in the web log registers, and so a sessionization process can use the keywords executed during the user's session. Such web mining algorithms can be applied to extract significant patterns about the users information needs. This process is known as "*Web Query Mining*" (WQM) [16, 15] and traditionally it has been applied by big search engines, like Google, Yahoo!, etc., to improve their page indexation and search process [17]. While WQM might be most useful in larger search engines, it is being applied in specific web site.search engines also.

It has been demonstrated that a locally applied WQM can show valuable patterns about the web user information preferences and which can then be extracted and used to improve the site's structure and content [247]. These keywords become a valuable resource for improving web page text content when they are to be revised or updated. Additionally keywords can be considered as hyperlink information for pointers to others pages [71].

5.3.6.2 Prefetching and caching

A web site is a collection of files administrated by a web server, which is a software running on an operation system. The main objective of the web server is to provide the web objects required by a web browser. Depending of several factors, such as the network bandwidth, the capacities of servers and clients, the web user can perceive latency in web object retrievals. Latency slows user web browsing, limiting his or her normal operations at any web site, independent of the site structure and content.

To reduce latency, web resource caching has become an important part of network infrastructure [70]. Caching consists in creating and maintaining a repository for the web objects that users have required. If a user requires a web object which is in the caching repository, it can be delivered immediately; if not, the web object is retrieved from the web and then stored in the caching repository. This technique is successful when the required page is inside the caching repository. On the other hand,

prefetching [174] is based on the prediction of which web objects might be required by users for downloading and caching them. If the prediction is wrong, the latency problem will be exacerbated with many web pages downloaded, using up network bandwidth, but not required by the user.

In practice, caching is implemented by using a network proxy server, which is a software running on a computer with storage capacities, and where each web object retrieved is stored and indexed, for further user requests. Thus, if several users ask for pages belonging to the same web site latency is reduced as the pages are already stored on the proxy server. Proxy effectiveness depends on index quality, i.e., if the web pages required are in the index, the user will receive them quickly. So an efficient prefetching algorithm is a key factor.

There are several approaches for predicting which pages might be requested by the users. In [79], Markov techniques are applied for recognizing access patterns in the user page request behavior, and in [140] a probabilistic model is applied for simulating the user request. Other approaches predict by analyzing the most often requested pages [29] and the associated statistics between the pages and the related objects. Most advanced prediction tools use the hyperlink structure inside of the requested object for performing the prediction activity, as is the case with the commercial products Web Collector³ and PeakJet2000⁴.

On the other hand, web mining techniques have also been used for predicting web page tasks. [10]. Generally, the algorithms employed attempt to discover the common accessed page sequences and so predict the next pages to be requested. Association rules have been used to construct an n-gram model to predict the next pages that are to be requested [253]. By using the web logs registers in the proxy server, the users original page access sequences are reconstructed and the frequent sequences are extracted by implementing a page prediction algorithm [251]. An interesting development uses proxy server web logs registers, of the web page hyperlink structure

³<http://www.inkey.com/save30/>

⁴<http://www.peak.com/peakjet2long.html>

of the retrieved pages for discovering common access sequences [69] to predict the next requested pages and by using their embedded hyperlinks, the other related web objects.

Latency will continue to be a problem, independent of network bandwidth and computer capacities, as new applications with more web objects require greater resources. So research on prefetching and caching is a fertile field for applying web mining algorithms that complement traditional techniques like statistics and probabilistic models.

5.3.6.3 Helping the user's navigation in a web site

Very often, when the user is looking for information in a web site, he or she may not be able to find it, even though the information is present in these pages. This situation happens because the web site structure is incorrect, the pages with relevant information can not be found easily, the page content is wrongly distributed etc. This situation can be improved, in part, by helping the user navigate the site, i.e., making recommendations or information hints that helps the information search task.

Navigation recommendations can be developed from user browsing patterns, (described above). With association rules and sequence patterns for predicting the user's next page [104, 106, 250], this information can be used to make navigation recommendations, by showing possible web page links sometimes accompanied by information hints that include an abstract of the suggested page.

Clustering techniques have been used for grouping similar user browsing behavior and extracting navigation patterns [83, 123]. Navigation directions can be recommended by assuming that if a user visits a similar sequence of pages described by the pattern, then he or she would be interested in other pages content identified by the pattern. As [163], the page sequences are the visited web objects URLs, are used as inputs for a clustering algorithm to extract navigation patterns. A similar process is applied in [195], where the pages visited by a user are considered as an independent

directed graph. If two graphs that belong to different users have similar sub-graphs, it means that the users' browsings behaviour was also similar.

A combination of web content and web usage data is used in [235]. User sessions are compared by the sequence of visited pages and the time spent on each of them, together with the page text content. Then, an artificial neural network is constructed to extract navigation patterns and used as inputs to a recommendation engine which provides navigation suggestions to users [243].

Web usage mining techniques are used mainly for understanding the user behavior on a particular web site and for providing navigation recommendations together with suggestions for changing the web site structure and content. These are part of the web personalization approaches to be examined in Chapter 6 .

5.3.6.4 Improving the web site structure and content

By using web logs registers and simple statistical methods, significant information about web page utilization can be extracted and used to change web site structure and content, for example, dropping a page with a small ratio of visits.

The process becomes more complex with web usage mining, where the algorithms are applied to discover patterns about user navigation and content preferences. This new knowledge is then used to change web site structure and content. Regarding the changes about structure, the techniques are concentrated on discovering user page visit patterns in a web site session. By using the patterns extracted by clustering techniques [62, 64, 195, 244] it is possible to change the web site structure [123, 243]. Another interesting approach is to discover common access paths [106] and association rules from sequence patterns and so modify hyperlinks [104].

Content changes have fundamental relationships with the free text used in web pages. While contents like pictures, movies and sounds could be more important in the future, today users are looking mainly for textual information by using search engines

or simply by web page inspection. So the method for creating the web pages [225] should examine a combination of words that generate text contents with significant information, based on the user preferences [43, 154]. In this sense, the utilization of key-text components, as it was shown in the section 5.2.3.3, could contribute to the improvement of the web site text content. These key-texts can be extracted assuming a correlation between the time that the user spends on a page and his or her interest in its content [242]. Then, feature vectors can be created and used as inputs for clustering algorithms, which extract patterns about significant pages for users and so identify the web site key-texts[190, 191]. In general, web usage mining techniques aim to personalize the web site structure and content for individual users [164], which is essential for new web-based systems.

5.3.6.5 Web-based adaptive systems

Adaptive web-based systems are predicted to be the next web development. WUM, amongst different techniques for adapting web-based systems is particularly promising due to its direct application on usage data[183]. Ideally, the adaptation task should be performed automatically on the web site. However, it is considered to be high risk, so that semi-automatic adaptation is the most common current approach. Moreover it is relatively straightforward as changes to the web site are the responsibility of the person in charge or web master. The PageGather system that construct index pages i.e., pages whose content are hyperlinks to a set of pages revolving around a similar topic is particularly valuable[179]. It uses a clustering technique for extracting clusters of pages from the web data generated during the user session. The resulting index pages are proposed for web site structural changes.

A proposal to increase user operation efficiency on a web site is a system for analyzing the web user operation activities and browsing behavior, [146]. Data is used as WUM algorithms inputs to extract sequence patterns to estimate the concatenation of hyperlinks between two pages or their elimination to allow the users to get to more interesting pages quickly.

Another approach is to use self-organizing feature maps to extract navigation patterns that visualize the sequence of most frequently visited pages. With this knowledge, web site hyperlinks modification can be suggested, such as the elimination of hyperlinks, new hyperlinks between and among pages, as well as changes to the page's position in the web site tree [233, 238].

5.4 Summary

Web mining is the application of data mining techniques to data originating on the Web, also known as web data. This requires the application of particular preprocessing and cleaning stages for transforming the web data into feature vectors and, in some cases, variations in the original data mining algorithms. Depending on the web data source, web mining techniques are classified as web structure mining (WSM), web content mining (WCM) and web usage mining (WUM).

Significant information can be extracted by using web structure mining, regarding the relevance of a web page or in a web community, i.e., a set of web sites sharing similar interest topics, with pages that point to other pages in the community. Page relevance is defined as the importance of a page in the web community or Web. A page is relevant if it is pointed to by other relevant pages. Analyzing a group of web sites to extract relevant pages provides information about the web site for the community and the Web in general. Page content can be classified as authoritative and which are hub. The former are natural information repositories, i.e., whose content have a high relevance for the web community and for the Web. The latter are pages that concentrate links to relevant pages, for example “my favorites web pages”.

To identify which pages are authoritative and hub, it is important that page's ranking is performed by a search engine, like Google or Yahoo!, which receive inquiries from users, (as a list of terms related to a topic), and then returns web pages that contain the terms. The question is, which pages should be shown to the user first?

- clearly those with the greatest relevance. The two most popular algorithms for ranking pages are the HITS and the PageRank.

In the HITS algorithm, the page ranking process is performed as follows. Supposing that we have a page repository that contains the n web pages and given the user query a set of m pages are retrieved, with $m \ll n$. Then m pages are ranked by the identification of which are authoritative and hub. The authoritative, with the highest relevance, are shown to the user. The process is executed online and depends on the terms of the query.

In PageRank, for the same n pages, page relevance is pre-calculated off-line, by using hyperlink information per page to extract how many times a page was indicated by another. This procedure is independent of the user's query and the page ranking known before any query. The method assumes that page relevance is the product of a consensus with the creators of other pages. The PageRank results do not show the real relevance of pages in the web community.

It is clear that the relevance of a page depends on the correct identification of pages that share a similar theme. So the identification of web communities improves the information search task by concentrating information about a particular study area on the Web.

Web content mining requires a data preprocessing stage to transform web page content into feature vectors. The vector space model is a common method by which it represents a document, in this case a web page, as a feature vector. Prior to working with web pages, it is necessary to apply some variations in the calculus of word weights and examine additional meta-information. In this way an entire web site or web community could be represented in a vector space. However, if a new page with different words is to be added it implies the recalculation of the weight in the entire set of vectors.

As the set of words used in the web site's construction could be enormous, it is

necessary to clean stop words and to apply a word stemming process. This process will reduce the vector dimensions and so improve the operation and performance of web mining algorithms on web data.

The content of a web page is different from a written document. In fact, a web page contains a semi-structured text, i.e., with tags that give additional information about the text component. Also, a page could contain pictures, sounds, movies, etc. Sometimes, the page text content is a short text or a set of unconnected words. This is a challenge to web page content analysis which can only be tackled by using web content mining algorithms.

Because the web logs form a complete register of the web user activities in a particular web site, it is possible to analyze user behavior for proposing changes to the site. Web usage mining techniques have been effective for the analysis of web log registers through the pattern extraction. Changes in the web site hyperlink structure can be implemented, by using the extracted navigation patterns, in order to help users search for information in the web site. An analysis of user preferences provides information about which text contents he or she is looking for and, with this information, changes to the web site page text content can be proposed.

Web usage mining can be grouped in two main categories: mining user navigation patterns, to improve the web site structure and content, and user modelling in adaptive interfaces, or “personalization”. The latter is key to the creation of the new generation of web-based platforms, which need personalized systems for improving the relationship with their users.

Chapter 6

Web-based personalization systems

*Nature is just enough; but men and women must
comprehend and accept her recommendations.*

Antoinette Brown Blackwell

A well-organized web site structure and content should help the users to find the information they are looking for. However, in practice, it is not always like that. Sometimes the web site structure is complex, hiding the information and causing a “*lost in hyperspace*” feeling to the user. On the other hand, when the web site contains a simple context, like free text only, it does not become attractive for users.

The above situation is inherited from the origins of the Web and thus the following question has always existed: “how can we prepare the correct web site structure and content at the correct moment for the correct user?” The answer is not simple and, for the moment, there are only approximations to a possible final solution. It seems that the key lies in comprehension of the user behavior in a web site, and using this knowledge to construct systems for personalizing the site for individual users.

This chapter describes the main personalization approaches for web-based systems used in the interaction with the users. Special attention will be paid to the

adaptive web sites and their contribution to the new portal generations.

6.1 Recommendation Systems

It is a common practice that every time we want to satisfy a desire or a need, we ask for help from a person that we consider more advanced and specialized in the subject area of interest. Consider the following situation in everyday life. A person has a health problem and needs to see a medic. One way is to search for information about medical practitioners by several means, e.g. Internet, newspapers, yellow pages, etc. Another usual way is to ask some friends for a recommendation about a good doctor or medical institution. This situation is very common, persons usually ask for recommendations, because the best way to avoid mistakes is by using the experience acquired by others.

We are constantly asking for recommendations for buying, eating, etc., and when the person or institution gives us good recommendations, a very special bond is created. Some experts call this “*creating customer loyalty*” [96]. When the business is small, it is not difficult to advise the customer with recommendations, but when the business is big or is growing, the number of assistants required for providing good advises to customers could exceed the physical capacities of the place where the business lies. Also, it would be economically counterproductive if the assistants have to attend people all day, including here companies with personal working in shifts. How to reduce the number of assistants, but support the customer queries? Again, the information technologies seem to offer the answer, by using pre-defined actions in front of a question, something like an artificial assistant.

Artificial recommender systems attend to emulate the human recommendation, tracking past facts performed for a group of persons, for instance products acquired, Frequently Asked Questions (in short FAQ), etc., for making new recommendations to an individual person.

Formally the recommendation problem can be expressed as follows [4]: Let $U = \{u_1, \dots, u_m\}$ be the set of all users and $I = \{i_1, \dots, i_n\}$ be the set of all possible items (books, CDs, DVDs, etc) to be recommended. Both sets depend on the business and could contain millions of elements. Let Γ be a function that measures the usefulness of the item i_k for the user u_j , i.e., $\Gamma : U \times I \rightarrow R$, with $R = \{r_1, \dots, r_l\}$ the set of nonnegative values for Γ function. Then

$$\forall u_j \in U, \quad I'_{u_j} = \arg \max_{i_k \in I} \Gamma(u_j, i_k), \quad (6.1)$$

is the set of items to be recommended to user u_j .

The Γ function depends on the recommendation system implementation, but it is usually represented by a rating, for instance, “*the most request products*” sorted list.

6.1.1 Short historical review

In the beginning of the seventies, some recommender systems were created under an emergent discipline known as Information Retrieval (IR). In these systems, the user asks for a document by writing a set of keywords, and the system shows the documents that contain the keywords (see the SMART system in [198, 197]). The rapid growth of number of documents in digital format made IR a fruitful research area in the coming years. A wide range of algorithms were developed that aimed to search for specific words and, sometimes, complex concepts in databases containing text documents. However a new question appeared: “From the entire set of documents, which of them contain the relevant information that the user is looking for?”. In IR, this problem has been tackled by using relevance ranking methods, which show how to retrieve documents ranked according to a certain criterion.

Other recommender approaches emerged in relation to the use of electronic mail, in short e-mail. Soon after e-mail had become an universal tool, persons and even

institutions began to use the e-mail for sending irrelevant information, also called as junk or spam e-mail, e.g. for product and service promotions. In general, these spam messages contained undesired text for the users. Some IR methods have been developed for automatically detecting and filtering junk e-mails and, in some cases, by consulting the final receiver user [74].

Information filtering has been a powerful tool for recommender systems development, because it behaves like a human being. That is, when we need a recommendation, in fact the recommender is filtering the entire set of information that he possesses to prepare the recommendation as a small set of sentences. It is clear that any information filter has a risk, because some important recommendations may not be selected. In this sense, the *collaborative filtering* [98] proposes that people collaborate to help others perform filtering by recording their responses to information that they have read. This was used in the seminal work *Tapestry* [188], an experimental mail system developed at the Xerox Palo Alto Research Center.

In essence, the Tapestry system works as follows. Let's suppose a user is looking for "*recommender system*" subject and performs a query by using as keywords "*recommender*" and "*system*". The result is a lot of documents, but after a quick user's view, only some of them are endorsed. Now the query and the selected documents are associated and the query will act as a filter provided for the user. By applying the Tapestry system on e-mails, the user will receive e-mails according to his own queries.

The basic idea on collaboration filtering has been used for the creation of several recommender systems, which in essence track the preferences and actions of a group of users for pattern extraction, and use this information to make useful recommendations for individual users [142]. For instance, in electronic book stores like Amazon.com and Bookpool.com, the users give their personal opinions about a book. This information allows the recommender system to rank the books and show them ranked when other users share the same requirement. There are more examples of recommender systems given in detail in [182, 188, 200].

In general, a recommender system considers three important aspects in its design [142]:

- Algorithms. Most of them are created by pattern extraction on user preferences.
- Human factor. It refers to the mechanics used for gathering the data related to the user preferences. Ideally, this process must be non-invasive for the user.
- Privacy issues. A very important thing is what happens with the data gathered from users. Because it is possible to get sensible data about a particular user, there is a consensus that the data must not be divulged. More privacy elements will be reviewed in the next sections.

6.1.2 Web-based recommender systems

In the beginning, the web-based recommender system have been seen only as a curiosity, but very quickly companies envisaged the potential of these tools for the purpose of increasing and retaining the number of virtual customers.

Because the interaction between a user and the web site is stored in the web log files, the recommender system has the necessary data for extracting user patterns and preferences, and for generating useful recommendations.

The recommender systems have proven their effectiveness in improving the web site and the user relationship which, from a practical point of view, means an increment in the company's sales and a major virtual market segment [200]. Of course, there were a lot of cases where the recommender systems did not work and the resulting in company losing both money and customers. However, it is a consensus thinking that the current web sites will need some kind of recommender systems for supporting the new user requirements and expectations.

In the traditional market, a store like a supermarket offers a limited amount of products for its customers. It is because the physical space and the local customer

preferences impose a very selective amount of products. In the digital market, this situation changes completely. Now, the customers are distributed around the world, and the physical space is only an old-fashion concept. Many companies that offer their products through the Web need to satisfy a wide demand, because the customer preferences can be completely different among them. What is explained above expresses a strong need in the digital market “*a customized product for each customer*” [184].

One of the most successful web-based recommender system was developed by Amazon.com. In the own words of its CEO, Jeff Bezos, “*if I have 3 millions customers on the Web, I should have 3 million stores on the Web*”. Amazon.com understood very quickly the need to develop systems to customize the virtual purchase, by using recommender tools.

The web-based recommender systems are mainly used in e-commerce for “*suggesting products to their customer and providing customer with information to help them decide which products to purchase*” [200]. The classic recommender system suggestion about a product include personalized information and an evaluation table that summarize the opinion of other customers that have bought the product in the past (collaborative filtering). A most advanced version of a recommender system also adds information about other complementary products (cross selling), in the style “*others customers that had bought X, also had bought Y and Z*”.

Today the recommender systems for e-commerce is an unquestionable need, because they allow to:

- Transform visitors in customers. Every day, a commercial web site receives a lot of visits. Some of them are performed by its customers and others by visitors that are looking for a product or service information. In some cases, the visitors can be a non-depreciable number of potential customers, who may be more valuable than the current web site’s customers. The question is how to transform a visitor in a customer?. A technique is to help the visitor to find

what they are looking for, through useful and personalized suggestions prepared by a recommender system.

- Increase the cross selling. When we visit a supermarket we usually bring with us the “*shopping list*” for buying. Also, it is common that the final purchase list contains items that we have not considered in the original list. It is because the supermarket logistic and product distribution have been organized for promoting the cross-selling between related products, for instance the bread near to the jam and eggs, such as a person that in the shopping list has only bread will consider to be a good idea to buy jam and eggs too. In the digital market the situation is similar. By tracking the customer preferences and purchase behaviour, it is possible to promote the cross-selling. A very good example can be found again in Amazon.com; when we are looking for a book, we automatically receive the book information and the recommendation about other related books, that have been bought together with the book of our interest.
- Building loyalty. In the digital market, the competition for acquiring new customers is hard. It is well know that the effort to catch a new customer is nearly five times more expensive that to retain a customer. Then the companies have developed mechanics for retaining customers by creating a value-added relationship. The loyalty construction is performed by a correct tracking of the purchase behaviour of valuable customers mainly. Some customers may not be profitable for the business overall. In the value-added process, recommender systems are used for planning the best strategy to tackle the customer preferences and to prepare an action to retain customers, by using well know methods like special promotions, discount, etc.

A good recommender system can improve the relationship between the customer and the company, through useful recommendations for acquiring the exact product and service that the customer is looking for. This practice is very important from the customers point of view, because it shows the company preoccupation for assisting

them. However, it is necessary to consider the privacy issues. A lot of badly directed recommendations can be considered an intrusion in the customer private life [66, 132].

6.1.2.1 Web recommender systems, particular approaches and examples

In early stages, the automatic recommender systems only performed simple database queries. However, due to the increase in hardware storage capacities and performance, it became possible to apply more complex data analysis methods, like data mining techniques. The first recommender systems used the nearest-neighbour and collaborative filtering algorithms [188] for predicting the product purchase decision and preparing the related recommendation.

In the PHOAKS (People Helping One Another Know Stuff) system [214], the collaboration filtering approach is applied on usenet messages for the creation of web resources recommendations. Another interesting approach was by using decision tree algorithms. This technique represents the pattern extracted from the input dataset in a tree model, where each branch represents a new decision for the user. Next, after few decisions, the user obtains the recommendation which is in the tree's leaf [256].

Traditionally, clustering techniques have been used in marketing for analyzing data containing user preferences, and for extracting significant patterns from the identified clusters. In the case of web-based recommender systems, the pattern extracted by using clustering techniques are used for preparing different kinds of recommendations, which can be grouped in online and offline recommendations. The former is mainly navigation recommendations for the user [163, 243] and the latter is straightforward recommendations for the web master for changing the structure and content of the web site [181, 191, 238, 240]. The above explained method for analyzing the user preferences, which demands a high amount of computer resources and is non-linear with the number of customers. This is an important fact to consider in a real world practical realization of a recommender system.

On the other hand, the item-based top-N recommendation algorithms (a complete

survey in [75]) focus on analyzing the similarities among various items for identifying similar items to be recommended. This process does not directly consider the user behavior in the web site, but generate item recommendations likely to be accepted by the user. Real-world successful cases of companies using top-N algorithms are Amazon.com, Book Matcher, Levi's Style Finder and My CD Now, among others [200].

6.2 Systems for personalization

The term personalization is nowadays widely spread in the e-commerce field as a new trend in the construction of web-based systems for supporting the web user requirements. Personalization can be defined as *“the ability to provide content and services tailored to individuals based on knowledge about their preferences and behavior”* [102]

From the beginning of marketing, people have been looking for ways of satisfying the desires and needs of a group of individuals. Initially, the techniques used aimed to configure the basic characteristics of a product or service to target a specific group of persons.

The development of new techniques for acquiring data about customer desires and needs, together with the advancement in the computer's capacities for massive data processing, have allowed to target the final group of persons to whom a product of service is directed. A natural evolution of the above process is One-to-One marketing [184], i.e., prepare a product or service for an individual; in other words, personalize the interaction between the business and the individual. In One-to-One marketing, the personalization is performed by understanding the customer individual desires and needs in order to focus marketing campaigns and pricing and distribution strategies for particular customers.

Personalization has succeeded in the low scale business, where the One-to-One marketing is an every day reality, for instance, in the small neighbourhood mini-

market, where the owner knows personal information about his customers, like *which kind of beer Mr. Simpson prefers* and inclusive *the amount of cans per week*. Then, when Mr. Simpson goes to the mini-market for his beers, the owner have the order semi-ready prepared, in this way creating a loyalty bound with the customer. Is it possible to extrapolate the above situation to big businesses like a supermarket with thousand of customers? Of course, by hand this task is impossible, but what if information technologies serve as a means for understanding the customer purchase behaviour?

The natural evolution is the personalization supported by computer systems for *“building customer loyalty by building a meaningful one-to-one relationship; by understanding the needs of each individual and help satisfying a goal that efficiently and knowledgeably addresses each individual’s need in a given context”* [189]

6.2.1 Computerized personalization

To apply One-to-One marketing to big scale businesses seems to be an *impossible task*, because it requires to acquire customer personal information regarding their desires and needs, which usually are confidential. Considering the magnitude of the business, the amount of information regarding certain customers could be very high. However, the current technology in massive data processing for behaviour pattern extraction (data mining) has shown a certain effectiveness in the understanding of the customer’s desires and needs [159], by rendering insight into the personal needs of individual customers possible.

In order to gather customer purchase behaviour and personal information, the companies adopt several techniques: the most common one is the *“point card”*, where an electronic card is given to the customer for collecting points in each purchase, which the customer can exchange for other products. This card allows the companies to track the customer purchase behaviour and its evolution in the time, because previous to giving the card, the company had required and received the customer’s

personal information.

In a digital market, in which the products and services are perceived, for the moment, only by descriptive information about their characteristics, personalization must focus on which information is displayed to the virtual customer. The main objective of personalization for information is “*to deliver information that is relevant to an individual or a group of individuals in the format and layout specified and in time intervals specified*” [127].

A delivery of informations can be the outcome of a direct question from the user, as what occurs in the search engines used today in e-commerce platforms. The concern is what happens when the search engine’s answer is ambiguous or includes irrelevant information for the user purposes? On the other hand, the delivering of information based on the user behaviour is also not exempted of problems. For instance, an excessive tracking and recommendations can be annoying for the user.

The computerized personalization is a complex matter with technical and non-technical challenges, which are well synthesized in [127]. The most critical of these challenges are:

1. Reduction of the irrelevant “*information hints*” returned by the personalization system. It is clear that if a user receives a huge set of information hints, it is very likely that he will be confused when deciding which hint is the most relevant one. The situation becomes worse when the hints are irrelevant in themselves.
2. Garbage in garbage out. If the data source to process for understanding the user behaviour in the Web is “*dirty*”, i.e., with errors, incomplete, etc., the results of any pattern extraction algorithm will not be accurate. As discussed in Section 2.2 about the nature of web data, we know that the noise in the data source can be reduced, but never completely eliminated. Hence, it is necessary to define mechanisms to store the web data for noise reduction purposes as well as methods for data pre-processing and cleaning.

3. Performance and scalability. It is not enough to construct a very profitable personalization system that provides interesting information hints. The time factor must be considered too, because if the personalization system needs 30 seconds or more to provide the hints, the users may have decided to visit other pages in the Web. Another point is the scalability. When a system grows with respect to the number of users, the effort to prepare the hints does not grow linearly. In fact, the effort may grow exponentially which can cause problems for the platform that support the personalization system.
4. Privacy invasion. Although the personalization system intention is to help the users, there may be psychological problems with respect to the user's feeling about the reception of information hints. The most important one is the *lost privacy feeling*, which means that the user thinks that he is being tracking and squeezing to get the maximum benefit for the company owning the personalization system.

Despite the above explained problems, the computerized personalization systems continued to being developed, especially in present days, when the web-based systems require tools to personalize the user session on the Web. In fact, web personalization is considered to be the new trend in the web site development [226].

6.2.2 Effectiveness of computerized personalization systems

To predict the effectiveness of a personalization system is a difficult task, because we do not know beforehand a priori how the information hints will be received by the user. In fact, it is beyond on the scope of any computer system to know the user's psyche at any moment of his life. Only some general approximations about what is commonly accepted [206] for the users could be applied to predict the effectiveness of the personalization system, but the real test will be performed when real users receive and evaluate the quality of the information hints [101, 243]

On the other hand, it is not clear if the personalized hints would be enough for delivering useful information to a particular user. In most cases, the user does not have well-defined preferences. A very specific list of hints can hide other relevant information and make it difficult for the personalization system to find out the user's desires and needs [246]. The use of personalization systems may incur inherent risks, especially in commercial transactions, where a poor personalization system may result in losing customers, which would be not well received by the companies.

Despite the above mentioned problems in making effective a priori personalization systems, most commercial platforms are using these kind of tools to improve the relationship with their customers. In the case of the Web, personalization became the new challenge for the development of effective commercial web sites.

6.2.3 Computerized personalization approaches

The computerized personalization is straightforward to provide information hints to the users. In e-commerce platforms, the personalization appears as a provider of personalized offering to one or more potential customers. Depending on the complexity of the personalization action, some platforms have developed a “*personalization engine*”, i.e., a computerized system for tracking the user behaviour which provide personalized recommendations, such as information hints.

In [3] , an excellent classification of the current approaches for personalization is presented, which distinguishes the following architectures: Provider-Centric, Consumer-Centric and Market-Centric.

The Provider-Centric architecture (see figure 6.1 part a) is maybe the most commonly used in the Web. In order to extract knowledge, the provider gathers information about the user behaviour, which will be used in the personalization action. The classic example of these approaches are the online shopping web sites.

indexPersonalization!Market-centric

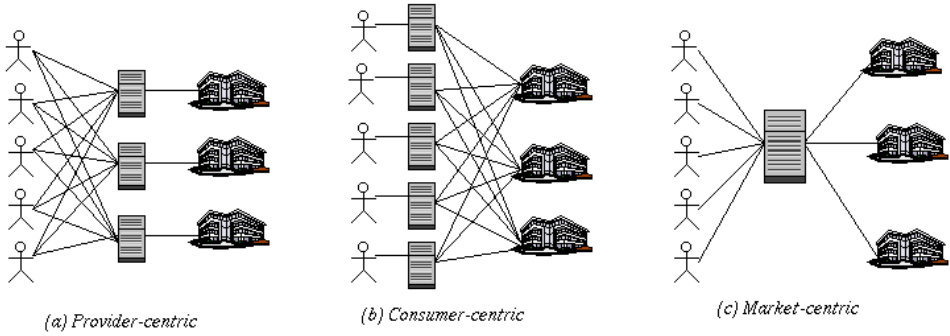


Figure 6.1: Web personalization approaches
Source: Based on Adomavicius [3]

The Consumer-Centric architecture is a software assistant for the user (see figure 6.1 part b). Whereas in the Provide-Centric approach the personalization action is aimed for increasing the provider benefit, in the Consumer-Centric approach, the personalization action purpose is to increase the consumer benefits. An example of this is e-Buckler [4], which provides personalized online shopping. This tool gathers information about its users and offers personalized information hints. For instance, if the user is looking for a shirt and thus puts the query “shirt” in the e-buckler, the tool will aggregate another previous collected user personal information such as the user shirt size and favourite colour, before executing a search on the Web.

Finally, in the Market-Centric approach (see figure 6.1 part c), the personalization engine works like an infomediary. Knowing the customer’s needs and the provider offerings, it performs a matching and prepares the information hints. An example of this approach is Hotels.com. This platform collects information from the web site of several hotels around the world, for instance, room’s price, hotel location, hotel’s agreements, etc. Then, when a user requires information about a room in a specific place, Hotel.com search in its database, selects the hotels with the best matching with the user requirements and returns a list with information about the selected hotels, usually ordered by price, but given to the user the possibility to sort the information

using another criterion.

The above approaches can be applied to develop offline and online personalization engines. However, in practice, the companies that use e-commerce platforms, are more interested in online systems, because they need to provide fast answers to their customers.

From the provider point of view, the personalization engine is used mainly for yielding information hints, products and service recommendations, e-mail campaign and cross selling products, etc. In general, these activities consider two important aspects [65]:

- Personalization of the presentation. It is related with the interface presented to the user, including the colours, position in the screen and fonts.
- Personalization of the content. This is the most complex part, where the information must be adapted to the particular user's needs.

In [65] a good analysis of computerized personalization engines is introduced for several kinds of final users, that include not only text content information needs, but also multimedia contents. In conclusion, the effort for creating personalized web-based systems is remarkable, and this action is called "*web personalization*".

In the next sections, we will review the main approaches and new trends in web personalization research.

6.3 Web personalization

In the literature, there are several definitions for web personalization. In [169], it is defined as "*how to provide users with what they want or need without requiring them to ask for it explicitly*".

More precisely, for the web-based systems implementation, the web personalization is “*any action that adapts information or services provided by a web site to the needs of a user or set of users, taking advantage of the knowledge gained from the user’s navigation behavior*” [83]. In other words, in [162], the web personalization is defined as “*the process to create web-based systems able of adapting to the needs and preferences of individual users*.”

From a practical point of view, the web personalization is the process where the web server and the related applications, mainly CGI-Bin¹, dynamically customize the content (pages, items, browsing recommendations, etc.) shown to the user, based on information about his/her behavior on a web site [155, 163]. This is different to another related concept called “customization”, where the visitor interacts with the web server using an interface to create his/her own web site, e.g., “My Banking Page” [225].

The key of web personalization is to understand the user’s desires and needs. It permits the design and construction of information repositories using the user transactions data, in order to predict the correct supply of products and services [23].

Personalization seeks to recognize patterns in the user behavior, in order to compare the patterns of new users and be able to make suggestions. A specific model about the user behavior and a measure that allow to compare two behaviors are required.

The personalization can be realized using the general methodology proposed in chapter 3, concerned with the Knowledge Discovery in Database (KDD) process. In fact it shows a clear way to create information repositories, make usermodels and extract knowledge from web data. Finally, the models and results should be checked by the experts and, by using their expertise about the business, the cycle is closed. Then, the personalization system can use the patterns and knowledge discovered.

¹Common Gateway Interface <http://www.msg.net/tutorial/cgi/>

6.3.1 Aspects of web personalization privacy

When constructing a system of web personalization, it is necessary to consider that each user has different information needs. Because it is impossible to perceive the user's thoughts, any personalization action is only an approximation of what we believe the user is looking for.

From previous sections, the common strategy for personalizing the interaction with the user is through information hints. These hints could be hyperlinks to other web objects (pages, documents, pictures, etc.) in web personalization. The hyperlink is accompanied by a brief text containing a simple abstract of the web site document content. Sometimes this abstract is prepared by hand and, in other cases, an automatic system obtains the abstract from the web site object text content. For example, Google.com, where by given a query containing keywords, the search engine returns the URLs with the web object that contains the keywords. For each URL, a short abstract with few sentences contained in the web object where the keyword appears is provided.

For preparing the personalization action, it is necessary to model both the web user behaviour and the web objects in the web site, in order to predict the web user information needs. This includes the modeling of web objects, like pages, site structure, user behavior, and categorize the objects as well as determine the actions to be recommended for personalization.

With respect to user behavior, it is necessary to consider that the users do not read a web page completely, but just scan it [172]. A web page must take into account the text that include keywords and bullet lists.

Some authors [59, 172] consider that users only look for things in their mind, and ignore aspects of design and contents. In other words, they are extremely "goal driven" and any personalization task should aim to help them to find what they are looking for. This must be done without causing confusions or making the user feel

that he/she “is lost in the hyperspace”.

Web personalization can be performed in two ways [162, 164]: off-line and on-line. In the first approach, the structure and the content are changed in the web site in order to help future users. The latter approach mainly corresponds to giving on-line navigation recommendations to the user [118].

Web personalization actions constitutes an iterative process [3] that starts with the understanding of the web user’s behaviour. This is achieved by gathering comprehensive information about the user, which is then used in the knowledge extraction task for storing user behaviour patterns. Next, the personalization engine uses the user behaviour patterns to prepare the personalized information hints. These can be directed to either individual or groups of users.

As the information hints are only a nice “*help intention*” before the recommendations are accepted by the user, it is necessary to measure the impact of the personalization action in the final user. This is a complicated matter and though there are some heuristic methods for testing the personalization action a priori, the real test is with real users, who may not be interested in giving their direct opinion about the quality of the information hints.

However, by using usability tests and the user’s reaction reflected in the web log files, it is possible to estimate the effectiveness of the personalization action. For instance, if a page is included in the information hints and the user follows this recommendation, then a register with this action is stored in the web log files. Otherwise, no action is stored in the web log files, given clear indication that the information hints have not been followed by the user.

Based on the effectiveness of the personalization action, the web objects and user behaviour models can be changed or adapted and sometimes it is necessary to redefine the user information to be collected. In the same way, the knowledge extraction algorithms could be changed, which take us again to the first stage of the

personalization process.

Because, usually, the information contained in a web site is often changed after a certain period of time, as it may not be relevant anymore for the purpose of that web site, the web personalization process never finishes. It is necessary to periodically perform the personalization actions.

6.3.2 Main approaches for web personalization

Personalization techniques have ceased to be a simple curiosity in web site design and became a powerful tool for participation in the digital market, reason for which many companies have invested large amounts of money in the implementation of personalized tools for their e-commerce platforms. This has been estimated to 2.6 billion of U.S. dollars in 2006².

A web personalization system can provide several levels of interaction with the user, from simple greetings messages to online recommendations. Considering the functionality, a good classification of the web personalization approaches has been presented in [183] and is given below:

Memorization. This is the most simple expression of personalization, and it consists in storing the basic information about the user, such as name and pages visited. When the user revisits the web site, the system recognizes him/her, showing the user's name and part of the last visit.

Guidance. It refers to assisting the user in order to find what he/she is looking for. In this sense, the personalization system can recommend links to other pages and related content.

Task performance support. It involves actions on behalf of the user, such as sending e-mails, complete queries and even, in advanced systems, represent the

²<http://www.datamonitor.com>

user interests, for example in a negotiation.

The common factor in any web personalization system is the need for having adequate mechanics to understand the user behavior in a web site. Web usage mining algorithms are the current techniques to achieve this. Although the variability within these tools may be large, we can classify the personalization approaches into three major categories [83]:

- Decision rule-based filtering. Here, the personalization tool applies a survey to the user, based on a decision tree, then, answer by answer, the user finally receive the information hints as a result.
- Collaborative filtering. As it was introduced in Section 6.1.2, the users are invited to rate the objects, for instance books, and give his opinion about them. The assumption is that an user with similar preferences would like to receive similar information about a product or a service.
- Content-based filtering. This group of techniques aim to discover personal user preferences by applying machine learning methods on the data collected from the user behaviour in the web. Then, based on the previous user behaviour, the tool return some information hints.

All of the above approaches make use of techniques for processing data originated in the web and, in this context, the web mining algorithms play an essential role [83, 176, 183].

6.3.3 Privacy aspects of web personalization privacy

Up to what point is a personalization engine is not invasive with respect to user privacy? It is a question beyond the technical aspects of the problem, but very

pertinent and important, as the privacy invasion feeling can be the major reason for which the users may not visit a web site with a personalization engine.

It is true that a web based personalization system help in the creation of a loyal relationship with the users, but it could be at the user's privacy expenses [132]. In order to know the user feeling about the privacy aspects in web personalization, in [66] opinion surveys have been carried out on real users . The results showed that the users appreciate the effort of helping them, but the majority of the users do not agree with the utilization of personal data and, even more, they do not like the fact that the companies share these data with their business partners.

By considering the origin of the user information in the web personalization, these data can be grouped in [132]:

- Explicit data. These are provided directly by the user, for instance, his age, name, nationality information, etc.
- Implicit data. These are inferred from the user behaviour; for instance, the visited pages, search queries, purchase history, etc.

Out of the above described types of data, which are public and which are private?. Of course, it depends on the legal definition, which is particular for every country. Some industrial countries have tackled the data privacy matter and implemented some regulations in the national laws. Such is the case of the U.S.A.³, where the privacy legislation covers very few types of data, but it is expected that this law will become stricter. Because, probably, each country has its own particular vision about the personal data privacy matter, it is quite difficult to reach a global law or norm. However, there are principles commonly accepted for data privacy, which can be applied in the web context.

³Self-Regulatory Principles for Online Preference Marketing by Network Advisers. Network Advertising Initiative, 2000

In the OECD Privacy Guidelines⁴, we find a set of norms about the personal data flow between countries. Some of them restrict the data flow to countries that do not provide enough levels of data protection. Similar norms exist in the European Data Protection Directives⁵, which define the minimum elements of the national privacy laws for the member states of the European Union.

The privacy laws can regulate the kind of protection for personal data. For instance, beneath the parsimony principle, the data collected for a specific purpose must be used only for that declared purpose. More specifically, the data gathering process must inform about the purpose of why the data are collected. Inspection of the data, in order to modify, block and even erase the data in case they are obsolete or there is suspicion of data privacy violation, should be allowed.

The above explained rights tend to be generic in any privacy matter. However, because the privacy laws are different in each country, and finally the privacy preferences depend on each individual user, it is impossible to design a web-based personalization system that can deal with all privacy requirements around the world. The current solution is to tailor the option of privacy to each user.

Today, the trend and best practices in web-based personalization systems tend to [132]:

1. Inform the user that he is about to enter in a personalized system, which will get data about his behaviour with the system, and tell how these data will be used.
2. Obtain user consent for yielding the personalization task. Some web-based systems show a previous window explaining in summary the personalization matters giving the user the possibility to explore for more information.

⁴Recommendation of the Council Concerning Guidelines Governing the Protection of Privacy and Transborder Flows of Personal Data., OECD, 1980

⁵Recommendation of the Council Concerning Guidelines Governing the Protection of Privacy and Transborder Flows of Personal Data., OECD, 1980

3. If it is technically possible, maintain several versions of the web-based system, some of them with a personalization engine and others without. The idea is that the user should select with which web-based system he wishes to work.
4. Provide an explanation about which security policies are being applied for maintaining sensitive personal user's data.

The above mentioned practices are only a minimal set of requirements for dealing with several important principles in privacy laws. Of course, there are laws that impose more severe requirements. In [132], the authors explain in detail the current efforts for providing stronger privacy policies.

6.4 Adaptive web-based systems

When a user enters a web site, the selected pages have a direct relation with the desired information that the user is looking for. The ideal structure of a web site should support the users in finding such information.

However, the reality is quite different. In many cases, the structure of the web site does not help in finding the desired information, although a page that contains it does exist [20]. Studying the user behavior is important in order to create more attractive contents, predict the user preferences and prepare links with suggestions, among others [165]. These research initiatives aim at facilitating web site navigation and, in the case of commercial web sites, at increasing market shares [150], transforming users into customers, increasing customers loyalty and predicting their preferences.

Since the creation of the web, researchers have been looking for friendlier ways of navigating web sites [34, 36, 163]. In early stages, objects with movements were used, e.g. Java Applets. Soon the dynamic web pages appeared and advances like the DHTML⁶ and the virtual reality were materialized. Because the main motivation

⁶Dynamic HTML, see <http://www.htmlcenter.com/tutorials/tutorials.cfm/116/dhtml/>

of a web user is searching for valuable information about a topic, some recommender systems facilities have been incorporated in the web site operation, for preparing information hints for the users. The next step in this evolution was the incorporation of elements for personalized Web such as personalization engines for personalizing the user navigation. However, the creation of specific web contents and structures in order to satisfy the visitors' desires is continuing to be a problem without a complete solution.

Nowadays a new generation of web sites is emerging, so called “*adaptive web sites*”, i.e., “*sites that automatically improve their organization and presentation by learning from user access patterns*” [181]. Several initiatives can be remarked [36, 67, 179, 227]. A consensus approach [126] is to combine artificial intelligence, user modelling and web mining algorithms in the creation of adaptive web sites.

The adaptive attribute suggests the ability of modifying the web site structure and the contents based on the individual user behavior. However, the most typical implementation is a gradual adaptation, e.g. a temporal variation of the text content or a change in the web page link, aiming at analyzing the variation of the user behavior after applying the changes.

6.4.1 A short introduction

A system is called adaptive if it changes its behavior by itself, using a user model for this. Figure 6.2 shows the classic loop “user modeling-adaptation” of adaptive systems.

The first stage is the collection of relevant data about the users. These data are mapped into a user model. It is the “user’s state mind” in the system. When a user interacts with the system, the adaptation effect is performed as an inference from the user model. It is important to emphasize that the adaptive system response depends on the quality of the information collected about the user, i.e., the system is a specific

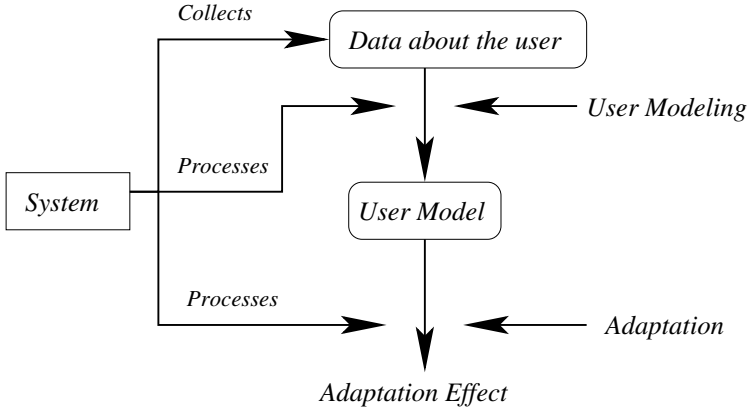


Figure 6.2: User model adaptation
Source: Brusilovsky [39]

application to solve a particular problem.

Since 1995, new applications of adaptive systems have been developed for the Web [40]. These are called “Adaptive Web-based systems” and their main issues are the adaptive navigation support and adaptive presentation (how to show a page content).

In the user interaction with the web, we find several motivations for developing of web-based adaptive systems:

- Different persons imply different users.
- The user behavior changes in time.
- When the user is searching for a specific information in the web, he/she could get the feeling of being “lost in the hyperspace”.

The adaptive web-based systems are being considered to be a solution to the above mentioned problems and aim to constitute the next portal generation.

6.4.2 Elements to take into account

The adaptive web sites have two kinds of potential users:

- Individual users. For the purpose of this book, they are identified as visitors and receive individual recommendations based on their personal interests.
- Web site operators. They are mainly web masters, that receive recommendations about offline changes in the web site.

The success or failure of an adaptive web site depends on the user's satisfaction. The challenge is not minor, because there are several aspects in the environment of adaptive web sites that may affect their feasibility and performance. Some of them are listed below:

Web data processing. In general, processing web data is not a trivial task. The main sources are the web log files, and these need to go through a preprocessing stage, firstly a **sessionization** process, secondly the cleaning of abnormal sessions (e.g. caused by web crawlers) and, finally, the design and construction of a specific algorithm to discover significant patterns.

Other sources like the web site structure and content can be considered too. The structure can show the relationship between internal and external pages (links to others web sites). This information is useful for understanding a web community.

In the case of content processing, the principal data is the free text within each page. However, the other objects (pictures, sounds, movies, etc.) are also of interest. The preprocessing stage is complex, because the outcome is a feature vector that should reflect the web content.

Measurement of success. A methodology to measure the effectiveness of the proposed changes is needed. A priori measures are still in their infancy, but some

theories have been developed [77]. It seems that the most effective method is to follow the recommendation given by the adaptive web site and see what happens. However, this is very risky for any institution, especially if the core business depends on the web site.

Real data sources. The user behavior in a web site can not be simulated by an automatic system, i.e., it is not possible to create web log registers using a simulation that makes sense.

A pattern discovery tool is necessary to be applied on data originated in a real web site. However, it is difficult to access real data from a commercial web site with a significant number of users, because usually this kind of data have associated strategic users. Usually, the researchers create experimental web sites in their own laboratories and generate data in the laboratory.

Impact in the user. The idea of introducing modifications in a web site is to generate changes in the user behavior. The changes may have a negative effect on the institution interests, e.g. if the user never visits the web site again.

Out of all possible modifications, the most riskier ones are the structural changes, i.e., adding or eliminating links, which may lead to the “lost-in-hyperspace” feeling situation.

6.4.3 Web site changes and recommendations

In creating a web site, a complex and meticulous process is followed, in order to get the best look and feel. Consequently, offline changes in its structure and content are not done every day, even in web sites prone to changes, such as newspapers web sites. Although every day the newspaper’s page change, the structure and the main themes follow an editorial line.

However, due to the dinamism of the Web and the e-commerce, there are periodic changes to be applied in the web site usually every month. Of course, these changes

must be deeply planned, and they are not exempt of risks, like losing customers.

The changes to perform in a web site correspond to structural and content adaptations. The structural ones include the addition and elimination of links. The content ones are mainly free text modifications, the variation of other objects like colors, pictures, etc., can be considered too.

Due to the risk of directly applying the changes proposed by an automatic system, the best way is to suggest recommendations for web users. Recommendations can be grouped in two categories: Online and Offline.

Offline recommendations are targeted to the web master or the person in charge of the web site. These refers to the addition or elimination of links, and changes in the web content. It is a non invasive scheme, because the web master can accept or reject the recommendations.

If off-line recommendations may look risky, the online ones are even more, because the user may lose the notion about “where he/she is” in the web site. In fact, some changes can violate the original web site links structure, i.e., missing physical link between the actual and the imposed page. When the user revisits the web site and wants to review the imposed page again, it may be hard for him to do so directly. In order to avoid the above problem, some authors [59] have proposed on-line recommendations that maintain the web site structure, i.e., there must exist a physical link between the actual and the recommended page. Those recommendations that break the structure should be considered for off-line changes. The online recommendations consists mainly in navigation suggestions showed at the bottom of the web page [40]. It is a non invasive scheme which provides to the user with the possibility of following the suggestion or not.

6.4.4 Adaptive systems for web sites

The web-based system adaptation to the user needs and desires can be tackled from two points of view [139]: adaptation by making recommendations to the user and reorganization of the web site as an adaptation to the user's interest.

In the first group, some commercial web sites have incorporated the *path predictions* [34], i.e., guessing where the user wants to go in the site and giving a recommendation about it. An interesting development is the WebWatcher⁷ [121], which uses a user's model in order to predict what link the user will follow from a particular web page. Before entering the web site, the users are asked, in broad terms, about what they want. When they depart the site, they are asked if they had found what they were searching for. Using the user's answers and the sequence of pages visited, the user's model is improved.

Clustering techniques have been successfully applied for implementing both kinds of adaptations. In [163] a clustering algorithm is applied on the web logs registers originated from a particular web site, for extracting the overlapping user preference profiles, by detecting clusters of URLs with the user's motivation in the site. This approach allows to prepare recommendations of likely useful links to the user and also modify the web site structure.

In [235], the authors used a Self Organizing Feature Map for extracting clusters about the user browsing behaviour and preferences, by using a similarity measure that combine the pages visited sequence, the page text content and the time spent on each of them during the user's session. Next, with the cluster identified, online navigation recommendations are shown to the user [243]. The extracted clusters also allow to prepare web site structure modification recommendations [233]; for instance, to change a hyperlink by using different colours, fonts or adding a new object like an icon and to connect unlinked web injects that are related based on the new knowledge extracted from the clusters [139].

⁷<http://www.cs.cmu.edu/~webwatcher/>

Also, by using clustering techniques, some web site offline content changes can be applied, e.g. in the case of web site keywords identification. The web site keyword is defined as “*word or possibly a set of words that is used by web users in their search process and characterizes the content of a given web page or web site*” [242]. By identification of the web site keywords, new text contents can be created in the web site.

Others adaptive web-based system approaches [181, 163] have proposed clustering techniques for predicting the user behavior. The **PageGather** algorithm [179] is an example of a conceptual clustering for index page synthesis. An index page is “*a page consisting of links to a set of pages that cover a particular topic*”. Then, the problem is to automatically generate the index pages for supporting the efficient navigation in the site.

Also, we can consider as web site adaptive solutions those implementations that help a user finds information on a particular site. For example, agent applications that adapt their behavior in response to the action of the individual visitors [178]. The agent’s goal is to give recommendations about related pages to users.

It is important to mention that adaptive concepts are also being applied in other Internet services [126], such as e-mails and news. For instance, the antispam systems clean the undesired e-mails that arrive to the mail server. Another example is the case of **SeAN** [12], a **S**erver for **A**daptive **N**ews. By construction, a news server operation is based on the sharing of newsgroups, some of them with irrelevant information for the users. The **SeAN** tools collect the interesting news for the user and propose an indexation for an easy reading.

6.5 Summary

Web personalization is a relatively new research area, but with a very rapid development, which have received a great attention from the academic and business com-

munities. Without being the unique reason for this fast growth, the main motivation is the institutional need for creating attractive web sites, to face the existing high competition in the digital market.

Although tremendous advances have been achieved, web personalization is continuing to be a hard and difficult problem. For the moment, the consensus is that any advance in the area needs to revolve around understanding the user behavior in the web. In this sense, the web mining algorithms are used in order to extract significant patterns from web data, which allow a better understanding of the user browsing behavior and preferences.

In the web personalization systems community, an ambitious project is considered to be the creation of good adaptive web sites, i.e., sites that change their structure and content on demand, based on the user behavior. It seems difficult to apply the changes directly, because the changes may be rejected by the user, which can create problems, e.g. loss of users. In conclusion, the system could give recommendations only.

Although there are some methods to a priori estimate the effectiveness of the proposed modifications in the web site, the final test takes place when the user faces the changes. This may incur considerable risks for the institution maintaining the web site.

Usually, the offline structure and content changes in a web site can be reviewed through an “usability test” where a group of selected users review the web site structure and fill a questionnaire with their impressions. The same methodology could be used with online recommendations, but it is not a real situation because the group of simulated users do not have an incentive to search for some topic and show a real user behavior. A real test needs real visitors, i.e., persons with a real intention of searching a topic. As mentioned above, the problem is that applying an online recommendation system directly over a real web site may be very risky for the business.

Chapter 7

Extracting patterns from user behavior in a web site

*Behavior is what a man does,
not what he thinks, feels, or believes*

Anonymous

A fundamental principle for achieving effective market participation is to understand the customer purchase behavior [137]. Marketing has developed powerful methodologies to capture information about current and potential consumers. To these can be added the new statistical techniques and data mining algorithms, with the objective of understanding consumer preferences and motivations about purchase behavior to a greater extent.

The common practice for collecting consumer behavior data is by opinion surveys about products or services. Surveys are generally based on direct inquiry on a sample or segment of a given population, called “focus group”. This technique is useful when potential or current customers are identified “face to face”. However, in virtual markets like e-commerce these techniques are not very effective, as virtual customers usually do not like to answer questionnaires. Nevertheless, by analogy, the Web may act as an online focus group survey - since the data related to the user

movement content preferences (web logs register) are collected for a particular web site, which allows the application of marketing theory and algorithms to understand user behavior.

Studying the user behavior is important in order to create more attractive content, to predict visitor/customer preferences [179], and to make links to suggestions, among others. Several research initiatives aim at facilitating web site navigation and – in the case of commercial sites – increasing market shares, transforming users into customers, increasing customers loyalty and predicting their preferences [229, 235]. In fact, the application of web mining tools allows the discovery of significant patterns within the web data. Each tool requires the cleaning of web data and the preparation of the specific input format; these are complex tasks, even when there are several tools or algorithms available.

The next section examines the main trends for modelling web user behaviour and the methods for extracting behaviour patterns.

7.1 Modelling the web user behavior

Analyzing user browsing behavior is the key to improving the content and the structure of a web site. By construction, each visit to a web site leaves behind important data about user behavior. The data is stored in web log files, which contain many registers. Some may hold irrelevant information so the analysis of user behavior is complex and time-consuming. [123].

Different techniques are applied to analyze web site user behavior ranging from simple web page use of statistics to complex web mining algorithms. In the last case, the research concentrates on predictions about which page the user will visit next and the information they are looking for.

Prior to the application of a web mining tool, the data related to web user behav-

ior has to be processed to create feature vectors, whose components will depend on the particular implementation of the mining algorithm to be used and the preference patterns to be extracted.

The majority of the web user behaviour models examine the sequence of pages visited to create a feature vector that represents the web user's browsing profile in a web site. In [252], given a web site S and a group of users $U = \{u_1, \dots, u_m\}$ who visit a set of pages $P = \{p_1, \dots, p_n\}$ in a certain period of time, the feature vector is characterized by a usage function $use(p_i, u_j)$ that associates a usage value between a page p_i and a user u_j , such as

$$use(p_i, u_j) = \begin{cases} 1 & \text{if } p_i \text{ has been visited by } u_j \\ 0 & \text{otherwise,} \end{cases}$$

The feature vector is $v = [use(p_1, u_k), \dots, use(p_n, u_k)]$ for $k = 1, \dots, m$.

Along the same lines, in [123], each URL in the site is designed by a unique number $j \in \{1, \dots, N_U\}$, where N_U is the total amount of real URLs. A user session is represented as $v = [s_j^{(1)}, \dots, s_j^{(N_U)}]$ where

$$s_j^{(i)} = \begin{cases} 1 & \text{if the user visited the } j^{th} \text{ URL during the } i^{th} \text{ session} \\ 0 & \text{otherwise.} \end{cases}$$

More advanced models also consider the time spent in each page visited by the user. In [164], the entire web site is considered as a set of URLs $U = \{url_1, \dots, url_n\}$, and the users' transactions is the set $T = \{t_1, \dots, t_m\}$. Then, for the user " i ", the associated transactions are $t_i \in T/t_i = \{u_1^{t_1}, \dots, u_n^{t_n}\}$ where

$$u_j^{t_i} = \begin{cases} 1 & \text{if } url_j \in t_i \\ 0 & \text{otherwise.} \end{cases}$$

The result is a vector of "0" or "1" that represents the URLs visited per user.

In [63], $t = \langle ip_t, uid_t, \{(l_1^t.url, l_1^t.time), \dots, (l_m^t.url, l_m^t.time)\} \rangle$ is a general user transaction where ip_t is the IP address, uid_t a user identifier, $l_i^t.url$ is the url of i^{th} page visited and $l_i^t.time$ the time stamp of the transaction " t ". A similar approach is used in [250], where the term "User Transaction" is defined by the set of URLs visited

and the time spent in each of them during the user session. Here the feature vector is $U = [(URL_1, d_1), \dots, (URL_n, d_n)]$, with d_j the time spent on URL_j .

These models analyze web user browsing behavior at a web site by applying algorithms to extract browsing patterns. A next step is to examine user preferences, defined as the web page content preferred by the user; and it is the text content that captures special attention, since it is used to find interesting information related to a particular topic by search engine. Hence, it is necessary to include a new variable as part of the web user behavior feature vector - information about the content of each web page visited. In [233], the user behavior vector analyzes the user browsing behavior and his/her text preferences.

Definition 1 (User Behavior Vector (UBV)) *It is a vector*

$v = [(p_1, t_1) \dots (p_n, t_n)]$, where (p_i, t_i) are the parameters that represent the i^{th} page from a visit and the percentage of time spent on it in the session, respectively. In this expression, p_i is the page identifier.

In Definition 1, the user behavior in a web site is characterized by:

1. Page sequence; the sequence of pages visited and registered in the web log files. If the user returns to a page stored in the browser cache, this action may not be registered.
2. Page content; represents page content, which can be free text, images, sounds, etc. For the purposes of this book, the free text is mainly used to represent the page.
3. Spent time; time spent by the user in each page. From the data, the percentage of time spent in each page during the user session can be directly calculated.

Figure 7.1 shows an eight page web site, where the user browsing data is stored in the associated web log file. After a session reconstruction process, the visitor's

usage data associated to the IP address **1.2.3.4** allows to create the vector $v_1 = [(1, 3), (2, 40), (6, 5), (5, 16), (8, 15)]$.

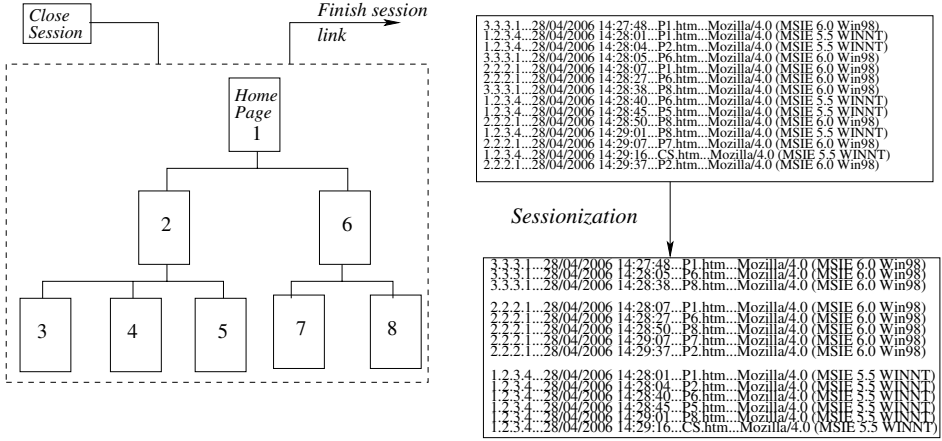


Figure 7.1: User behavior vector's creation

As the time-stamp parameter in the web log only shows the exact moment when the web object is requested, the time spent by the user in each page visited is calculated by the difference in time between the visit to two consecutive pages at the same session. This calculus can be problematic for the final page visited, as the user may have left the site and this cannot be determined exactly. There are some solutions to this problem. First, apply the average time spent at other pages visited during the user session; second, set up a maximum duration with the user in “inactivity status” after which it is assumed to be “finished session” status. For example if there is a security protocol for accessing a web site and there is no interaction between the user and the site for a determined period, then the site finishes the session, by sending a special page to the user as shown in Figure 7.1, (on the right side). It is helpful to examine the “finish session link”, i.e., a link used voluntarily to finish this session by the user, (see Figure 7.1, with the page “close session”).

Now it is clear that the length of a session depends on user browsing navigation - and this generates vectors of different lengths. Some mining algorithms require vectors with the same cardinality, and then the UBV will need to set the length, i.e., the n value. One solution is to set n to the average number of pages visited by the users. If so, the vectors whose length is less than n are classified as having null values; if not, vectors will be trimmed to the first n values in the user session.

The UBV can be used as an input of web mining algorithms, preferentially those related to clustering and classification techniques. In both cases, the comparison between vectors is essential, i.e., it is necessary to have a similarity measure that shows how similar or different two vectors are compare to the three elements - page sequence, page content and time spent - that form a UBV.

7.2 Web data preparation process

Of all available web data, the most relevant for the analysis of user browsing behavior and preferences, are the web log registers and the web pages [229]. The web log registers contain information about the page navigation sequence and the time spent at each page visited, by applying the **sessionization** process. The web page source is the web site itself. Each web page is defined by its content, in particular free text. To study user behavior both data sources - web logs and web pages - have be prepared by using filters and by estimating real user sessions. The preprocessing stage involves, first, a cleaning process and, second, the creation of the feature vectors as an input of the web mining algorithms, within a structure defined by the patterns sought.

Among these web mining techniques, special attention should be paid to the clustering algorithms. The assumption is that, given a set of clusters extracted from data generated during former user sessions in the web site, it is possible to predict future user behavior by comparing the current user session with these clusters and select the closest one. The information content of the cluster selected would be sufficient to

extrapolate either a future page that the user might visit or the content that he or she is looking for [163, 195, 243].

A key feature of clustering techniques is the similarity or distortion measure for comparing user behavior vectors. Vector components need preprocessing to transform them into relevant comparable information. Similarities must consider the efficiency of the web data processing. The calculus should be simple, but provide a clear idea about the similarities or differences of user behavior vectors. Similarity measures are the subject of the next section.

7.2.1 Comparing web page contents

There are several methods for comparing the content of two web pages, here considered as the free text inside the web pages. The common process is to match the terms that make up the free text, for instance, by applying a word comparison process. A more complex analysis includes semantic information contained in the free text and involves an approximate term comparing task as well.

Semantic information is easier to extract when documentation includes additional information about the text content, e.g., market language tags. Some web pages allow document comparison by using the structural information contained in HTML tags, although with restrictions. This method is used in [218] for comparing pages written in different languages with similar HTML structure. The comparison is enriched by applying a text content matching process [219], which considers a translation task to be completed first. The method is highly effective when the same language is used in the pages under comparison. A short survey of algorithms for comparing documents by using structural similarities is found in [42].

Comparisons are made by a function that returns a numerical value showing the similarities or differences between two web pages. This function can be used in the web mining algorithm to process a set of web pages, which might belong to a web

community or an isolated web site. The comparison method must consider efficiency criteria in the web page content processing [124]. Here the vector space model [27], introduced in section 2.3.1, allows a simple vectorial representation of the web pages and, by using distance for comparing vectors, provides a measure of the differences or similarities between the pages.

Web pages must be cleaned before transforming them into vectors, both to reduce the number of words - not all words have the same weight - and make the process more efficient. Thus, the process must consider the following types of words:

- HTML Tags. In general, these must be cleaned. However, the information contained in each tag can be used to identify important words in the context of the page. For instance, the <title> tag marks the web page central theme, i.e., gives an approximate notion of the semantic meaning of the word and, is included in the vector representation of the page.
- Stop words (e.g. pronouns, prepositions, conjunctions, etc.)
- Word stems. After applying a word suffix removal process (word stemming [185]), we get the word root or stem.

For vector representation purposes, let R be the total number of different words and Q be the number of pages in the web site. A vectorial representation of the set of pages is a matrix M of size $R \times Q$,

$$M = (m_{ij}), \quad i = 1, \dots, R \quad \text{and} \quad j = 1, \dots, Q, \quad (7.1)$$

where m_{ij} is the weight of word i in page j .

Based on *tfidf-weighting* introduced in the equation (2.2), the m_{ij} weights are estimated as,

$$m_{ij} = f_{ij}(1 + sw(i)) * \log\left(\frac{Q}{n_i}\right). \quad (7.2)$$

Here, f_{ij} is the number of occurrences of word i in page j and n_i is the total number of times that the word i appears in the entire web site. Additionally, a words importance is augmented by the identification of special words, which correspond to terms in the web page that are more important than others, for example, marked words (using HTML tags), words used by the user in search of information and, in general, words that imply the desires and the needs of the users. The importance of special words is stored in the array sw of dimension R , where $sw(i)$ represents an additional weight for the i^{th} word.

The array sw allows the vector space model to include ideas about the semantic information contained in the web page text content by the identification of special words. Fig. 7.2, shows the special words detection for marked words using HTML tags.

In the vectorial representation, each column in the matrix M is a web page. For instance the k^{th} column m_{ik} with $i = 1, \dots, R$ is the “ k^{th} ” page in the entire set of pages.

Definition 2 (Word Page Vector) *It is a vector*

$\mathbf{WP}^k = (wp_1^k, \dots, wp_R^k) = (m_{1k}, \dots, m_{Rk})$, $k = 1, \dots, Q$, *is the vectorial representation of the k^{th} page in the set of pages under analysis.*

With the web pages in vectorial representation, it is possible to use a distance measure for comparing text contents. The common distance is the angle cosine calculated as

$$dp(WP^i, WP^j) = \frac{\sum_{k=1}^R wp_k^i wp_k^j}{\sqrt{\sum_{k=1}^R (wp_k^i)^2} \sqrt{\sum_{k=1}^R (wp_k^j)^2}}. \quad (7.3)$$

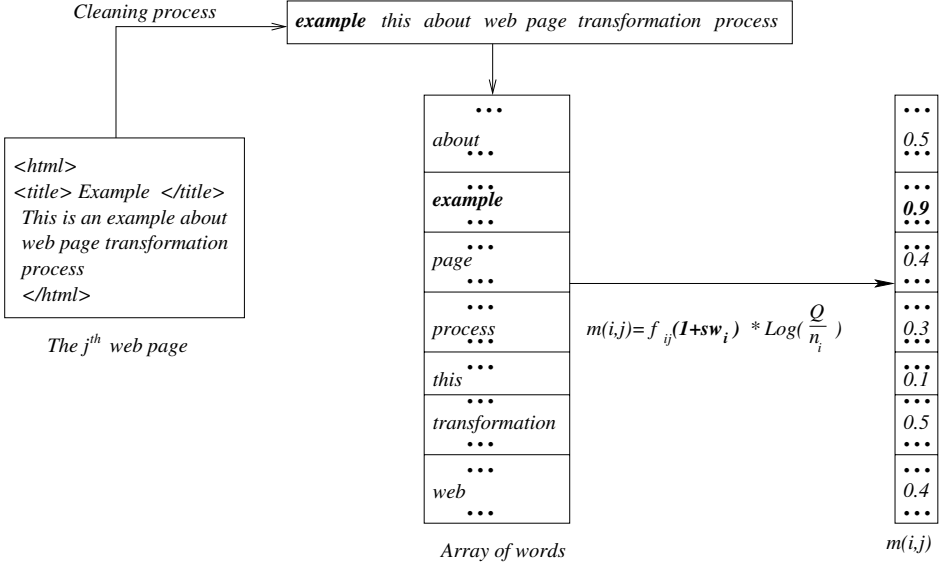


Figure 7.2: Including the importance of special words in the vector space model

The Eq. (7.3) allows to compare the content of two web pages, returning a numerical value between $[0, 1]$. When the pages are totally different, $dp = 0$, and when they are the same, $dp = 1$. Another important aspect is that the Eq. (7.3) complies with the requirement of being computationally efficient, which makes it appropriate to be used in web mining algorithms.

7.2.2 Comparing the user navigation sequences

After performing the sessionization task, an approximate set of pages visited by the user can be reconstructed. This navigation sequence can be illustrated by a graph G , as shown in figure 7.3, where each edge (web page) is represented by an identification number. Let $E(G)$ be the set of edges in the graph G . Figure 7.3 shows the structure of a simple web site. Assuming that two users have visited the site and the respective sub-graphs are $G_1 = \{1 \rightarrow 2, 2 \rightarrow 6, 2 \rightarrow 5, 5 \rightarrow 8\}$ and $G_2 = \{1 \rightarrow 3, 3 \rightarrow 6, 3 \rightarrow$

7}. Then $E(G_1) = \{1, 2, 5, 6, 8\}$ and $E(G_2) = \{1, 3, 6, 7\}$ with $\|E(G_1)\| = 5$ and $\|E(G_2)\| = 4$, respectively. In this example, page 4 has not been visited.

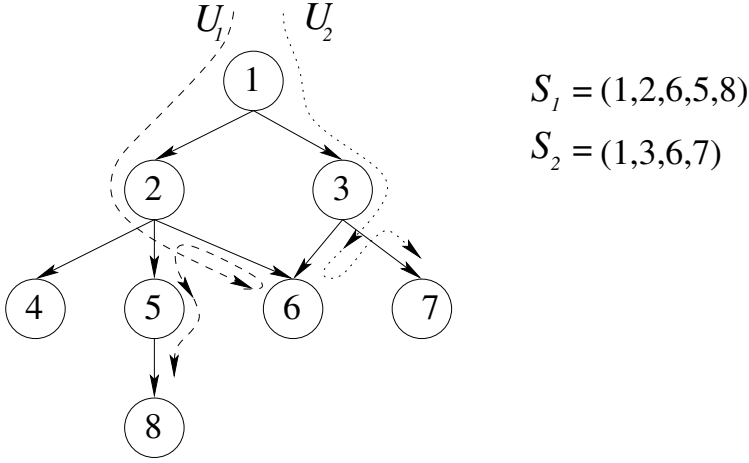


Figure 7.3: A web site with two navigation sequences

From the subgraphs G_1 and G_2 , the user navigation sequence $S_1 = S(G_1) = (1, 2, 6, 5, 8)$ and $S_2 = S(G_2) = (1, 3, 6, 7)$ are derived, respectively. They represent the visited pages and their sequence. Notice that, although both users go back to a visited page – page 2 for U_1 and page 3 for U_2 – if this information is not registered in the web logs, because the pages are in the web browsers cache, then the pages were not specifically requested again during the user session.

Different user sessions can be compared by a similarity measure of respective navigation sequences. The measure must consider how much two sub-sequences have in common. Equation 7.4 introduces a simple way to compare two navigation sequences [195]

$$dG(G_1, G_2) = 2 \frac{\|E(G_1) \cap E(G_2)\|}{\|E(G_1)\| + \|E(G_2)\|}. \quad (7.4)$$

Notice that $dG \in [0, 1]$ and if $G_1 = G_2$, then $dG(G_1, G_2) = 1$. In the case of disjoint graphs, $dG(G_1, G_2) = 0$ because $\|E(G_1) \cap E(G_2)\| = 0$.

, However, the equation 7.4 does not take into account the sequence of visited pages in each sub-graph; it just looks at the set of visited pages. In order to compare sequences, instead of calculating $\|E(G_1) \cap E(G_2)\|$, the node order must be examined. For example in Figure 7.3, both sequences can be represented by a string of tokens [195], such as $S_1 = "12658"$ and $S_2 = "1367"$ and the task is to know how similar both strings are. For this purpose, the *Levenshtein distance* [148], also known as *edit distance*, is used. It determines the number of necessary transformations to convert S_1 into S_2 . This distance can be used as a dissimilarity measure between sequences.

Given two sequences, $\hat{x}_p = (x_1, \dots, x_p)$ and $\hat{y}_q = (y_1, \dots, y_q)$, the Levenshtein distance is defined as:

$$L(\hat{x}_p, \hat{y}_q) = \begin{cases} p & q = 0 \\ q & p = 0 \\ \min\{L(x_{p-1}, \hat{y}_q) + 1, & \text{else} \\ L(\hat{x}_p, y_{q-1}) + 1, \\ L(x_{p-1}, y_{q-1}) + z(x_p, y_q)\} & \end{cases} \quad (7.5)$$

where

$$z(i, j) = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{otherwise.} \end{cases} \quad (7.6)$$

The definition of the Levenshtein distance conveys the importance of the order of the tokens to be compared. For instance, in order to compare $S_1 = 12658$ and $S_2 = 1367$, three transformations are required, but for $S_3 = "12856"$ and $S_4 = "1367"$ four transformations are needed. This characteristic makes the Levenshtein distance very suitable to compare two navigation sequences. Then, the final expression for dG is

$$dG(G_1, G_2) = 1 - 2 \frac{L(S_1, S_2)}{\|E(G_1)\| + \|E(G_2)\|}. \quad (7.7)$$

Using the example in Figure 7.3 and Equation 7.7, it leads to $dG(G_1, G_2) = 0.\bar{4}$.

7.3 Extracting user browsing preferences

Clustering algorithms is one of the most popular techniques to extract navigation patterns from the UBV. The first step is to define a similarity measure to compare UBVs. The algorithm must maintain the continuity of the user navigation in the site. At most web sites, users generally access a page, follow the current page's structured hyperlinks, rather than jump to the site's inner structure without following a hyperlink.

7.3.1 Comparing user browsing behavior

Let α and β be two user behavior vectors with cardinality C^α and C^β , respectively, and $\Gamma(\cdot)$ be a function that applied over α or β returns the respective navigation sequence. The proposed similarity measure is as follows:

$$sm(\alpha, \beta) = dG(\Gamma(\alpha), \Gamma(\beta)) \frac{1}{\eta} \sum_{k=1}^{\eta} \tau_k * dp(p_{\alpha,k}^h, p_{\beta,k}^h) \quad (7.8)$$

where $\eta = \min\{C^\alpha, C^\beta\}$, and $dp(p_{\alpha,k}^h, p_{\beta,k}^h)$ is the similarity measure between the k^{th} page of vector α and the k^{th} page of vector β .

The first element of equation 7.8 is the similarity sequence introduced in equation 7.7. The term $\tau_k = \min\{\frac{t_k^\alpha}{t_k^\beta}, \frac{t_k^\beta}{t_k^\alpha}\}$ is an indicator of the user's interest in the pages visited. The time spent on a page is assumed to be proportional to the interest of the user in its content. If the time spent by users α and β on the k^{th} page visited (t_k^α, t_k^β , respectively) is about the same, the value of τ_k will be close to **1**. In the opposite case, it will be close to **0**. This reasoning is supported by the following facts. Users can be grouped in two classes: amateur and experienced. The former corresponds to

persons unfamiliar with a particular web site and probably with web technology skills [227, 257]. Their browsing behavior is erratic and often they do not find what they are looking for. Experienced users are familiar with this or similar sites and have web technology skills. They tend to spend little time in low interest pages and concentrate on the pages they are looking for and where they spend a significant amount of time. As amateurs gain skills they slowly become experienced users, and spend more time on pages that interest them.

The third element, dp , measures the similarity of the k^{th} page visited. Two users may visit different web pages in the web site, but with similar contents, e.g., one page may contain information about classic rock and the other about progressive rock. In both cases, the users are interested in music, specifically in rock. This is a variation of the approach proposed in [123], where only the user's path, not the content of each page was considered.

Finally, in equation 7.8, the content of the visited pages multiplied by the time spent on each of the page is combined. So two users who had visited similar pages but spent different times on them can be distinguished as users who spend the same time on pages but with different contents.

7.3.2 Applying a clustering algorithm for extracting navigation patterns

Models of neural networks explain learning by generating clusters. An artificial neural network of the Kohonen type (Self-organizing Feature Map; SOFM), was chosen to mine user behavior vectors and to discover knowledge [205] about the user preferences in an unsupervised way. Schematically, the SOFM's output is represented as a two-dimensional array of neurons. Each neuron is constituted by an n -dimensional vector, whose components are called synaptic weights. In the SOFM, by definition, all the neurons receive the same input at the same time.

The SOFM's learning process is conceived as a network where the nearest neuron in the network to the presented example (center of excitation, winner neuron) can be calculated by a metric. Next, the weights of the winner neuron and those of its neighbors are updated.

This type of learning is called **unsupervised**, because the neurons “move” towards the centers of the groups of examples that they are trying to represent. This process was proposed by Kohonen, in his *Algorithm of training of Self Organizing networks* [135].

The topology defines the notion of neighborhood among the neurons. In our case, the toroidal topology is used[245, 229], which means that the neurons at the top edge are located next to the ones at the bottom edge, and the same for lateral edges (see figure 7.4). This topology respects the continuity of the user browsing in the web site [235], allowing a correct generation of clusters.

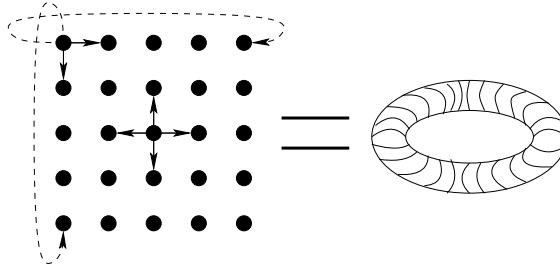


Figure 7.4: Neighborhood of neurons in a toroidal Kohonen network

The SOFM's operation requires vectors of the same cardinality, which is a configurable parameter for creating user behavior vectors. Let H be the number of elements in each vector. If a user session does not have at least H elements, it is not considered in the analysis. On the other hand, we only consider up to the H^{th} component of a session in the user behavior vector.

As previously introduced, the UBVs have two components for visited pages (the page identifier and the time spent on each page), so both must be updated for the winner neuron and its neighbors.

Let N be a neuron in the network and E the correct UBV presented to the network. The vector's time component is modified by a numerical adjustment, i.e., $t_{N,i+1} = t_{N,i} * f_t$ with $i = 1, \dots, H$ and f_t a time adjustment factor.

The page component uses a different updating scheme. Using the page distance, the difference between page content components is calculated as:

$$D_{NE} = [dp(p_{N,1}^h, p_{E,1}^h), \dots, dp(p_{N,H}^h, p_{E,H}^h)], \quad (7.9)$$

where dp is defined in (7.3).

In Equation 7.9, the vector components are numerical values representing the distance between pages. Then, the adjustment is made over the D_{NE} expression, i.e., $D'_{NE} = D_{NE} * f_\rho$, with f_ρ the adjustment factor. Hence, a set of pages must be found with distances to N are close to D'_{NE} . The final adjustment for the page component of the winner neuron and its neighbor neurons is given by equation 7.10,

$$p_{N,i+1} = \pi \in \Pi / D'_{NE,i} \approx dp(\pi, p_{N,i}^h) \quad (7.10)$$

where $\Pi = \{\pi_1, \dots, \pi_Q\}$ is the set of all pages in the web site, and $D'_{NE,i}$ is the i^{th} component of D'_{NE} . Given $D'_{NE,i}$, then the task is to find the page π in Π whose $dp(\pi, p_{N,i}^h)$ is the closest to $D'_{NE,i}$.

7.4 Extracting user web page content preferences

For many companies and/or institutions, it is no longer sufficient to have a web site and high quality products or services. What often makes the difference between e-business success and failure is the potential of the respective web site to attract and retain users. This potential depends on the site content, design, and technical aspects, such as the amount of time to download the pages from the web site to the user's web browser, among others. In terms of content, the words used in the free text of a web site pages are very important, as the majority of the users perform term-base queries in a search engine to find information on the Web. These queries are formed by keywords, i.e., a word or a set of words [145] that characterize the content of a given web page or web site.

The suitable utilization of words in the web page improves user appeal, helps effective information search, while attracting new users and retaining current users by continuous updating page text content. So the challenge is to identify which words are important for users. Most web site keywords are calculated from "most frequently used words". Some commercial tools¹ help identify target keywords that customers are likely to use while web searching [43]. In [17], keywords have been collected from a search engine, show global words preferences from a web community, but no details about particular web sites. A method for extracting keywords from a huge set of web pages is introduced in [43]. This technique is based on assigning weights to words, depending on their frequency in all documents. In this approach, a vector space processing is applied, i.e., cleaning of stop words and stemming reduction.

In [242, 239], the idea of "web site keywords" was introduced as *a word or a set of words that makes the web page more attractive for the user*. Assuming that there is a correlation between user interest and the maximum time spent per page, the analysis of word site keywords has taken two principal approaches:

¹see e.g. <http://www.goodkeywords.com/>

1. For experienced users, the spent time in a page has a direct relation with their interest, represented by particular words contained in the page.
2. The web site designer defines some words as “special”, since different fonts are used or a particular tag is applied, for example the `< title >` tag.

7.4.1 Comparing user text preferences

The aim is to determine the most important words at a given web site for users, by comparing the user text preferences, through the analysis of pages visited and the time spent on each of them [245]. It differs, however, from the previously mentioned approaches, as the exercise is to find the keywords that attract and retain users from the user web usage data available. The expectation is to involve current and past users in a continuous process of keywords determination.

User preferences about web content are identified by content comparison of pages visited, [245, 229, 233] by applying the vector space model to the web pages, with the variation proposed in section 7.2.1, Eq. (7.2). The main topics of interest can be found by using a distance measure among vectors (e.g. Euclidean distance),

From the user behavior vector (UBV), the most important pages are selected assuming that degree of importance is correlated to the percentage of time spent on each page. The UBV is sorted according to the percentage of total session time spent on each page. Then the ι most important pages, i.e. the first ι pages, are selected.

Definition 3 (Important Pages Vector) *It is a vector*

$\vartheta_\iota(v) = [(\rho_1, \tau_1), \dots, (\rho_\iota, \tau_\iota)]$, where (ρ_ι, τ_ι) is the component that represents the ι^{th} most important page and the percentage of time spent on it by session.

Let α and β be two user behavior vectors. The proposed similarity measure between the two important page vectors is introduced in equation 7.11 as:

$$st(\vartheta_\iota(\alpha), \vartheta_\iota(\beta)) = \frac{1}{\iota} \sum_{k=1}^{\iota} \min\left\{\frac{\tau_k^\alpha}{\tau_k^\beta}, \frac{\tau_k^\beta}{\tau_k^\alpha}\right\} * dp(\rho_k^\alpha, \rho_k^\beta) \quad (7.11)$$

The first element in (7.11) indicates the users interest in the visited pages. If the percentage of time spent by users α and β on the k^{th} page visited is close to each other, the value of the expression $\min\{\cdot, \cdot\}$ will be near **1**. In the extreme opposite case, it will be near **0**. The second element in (7.11) is dp , the distance between pages in the vectorial representation introduced in (7.3). In (7.11) the content of the most important pages is multiplied by the percentage of total time spent on each page. This allows pages with similar contents to be distinguished by different user interests.

7.4.2 Identifying web site keywords

In order to find groups of similar user sessions, a clustering algorithm can be applied [228]. The most important words for each cluster are identified by the identified cluster centroid.

Using the Eq. (7.2), a method to determine the most important keywords and their importance in each cluster is proposed. A measure (geometric mean) to calculate the importance of each word relative to each cluster is defined as:

$$kw[i] = \sqrt[\iota]{\prod_{p \in \zeta} m_{ip}} \quad (7.12)$$

where $i = 1, \dots, R$, kw is an array containing the weights for each word relative to a given cluster and ζ the set of pages representing this cluster. The most important word groups for each cluster can be selected by sorting kw .

The final number of different words are calculated using web page filters. The following procedure calculates the word weights, especially for special words sw_i (7.2). The most usual word sources, using examples from different types of web sites (eg.

call centers, banks etc.) are:

1. E-Mails. The offer to send user e-mails to the call center platform. The text sent is a source to identify the most recurrent words. Let $ew_i = \frac{w_{e-mail}^i}{TE}$ be the array of words contained in e-mails, which are also present in the web site, where w_{e-mail}^i is the frequency of the i^{th} word and TE is the total amount of words in the complete set of e-mails.
2. Marked words. Within a web page, there are words with special tags, such as a different font, e.g., italics, or a word belonging to the title. Let $mw_i = \frac{w_{marks}^i}{TM}$ be the array of marked words inside web pages, where w_{mark}^i is the frequency of the i^{th} word and TM is the total amount of words in the whole web site.
3. Asking words. A bank, for example, has a search engine through which the users can ask for specific subjects, by introducing key words. Let $aw_i = \frac{w_{ask}^i}{TA}$ be the array of words used by the user in the search engine and also contained in the web site, where w_{ask}^i is the frequency of the i^{th} word and TA is the total amount of words in the complete set.
4. Related web site. Usually a web site belongs within a market segment, in this case the financial institutions market. Then, it is possible to collect web site pages belonging to the other sites in the same market. Let $rw_i = \frac{w_{rws}^i}{RWS}$ be the array with the words used in the market web sites including the web site under study, where w_{rws}^i is the frequency of the i^{th} word and RWS is the total number of words in all web sites considered.

The final expression $sw_i = ew_i + mw_i + aw_i + rw_i$ is the simple sum of the weights described above. Then, it is possible to obtain the complete set of word page vectors and to calculate the distance among them, i.e., $D = (dp_{ij})$, with $i, j = 1, \dots, Q$.

7.5 Summary

To analyze the user behavior in a web site requires the modelling of user-site interactions. The user behavior vector (UBV) is central to this exercise as it contains the web data related to user navigation and preferences; that is the basic information about the sequence of visited pages, time spent in each one of them and their content. This information is the input to web mining algorithms for extracting user navigation and preferences patterns.

Prior to generating UBVs, the web site data must be cleaned and preprocessed. This task involves the sessionization of the web logs to reconstruct the users' session behaviour and transform the web page text content into feature vectors that represents the user text content interest. This a similarity measure to compare user navigation and content preferences has to be added, used in web mining algorithms to extract user behavior patterns.

Although there are many approaches for web mining, clustering techniques hold out the greatest potential for analyzing web site user behavior - in particular, the Self Organizing Feature Maps in thoroidal configuration, as it maintains user navigation continuity at the site. It is clear that other clustering techniques can also be used, although the methodology for extracting patterns out of the UBVs will be the same.

UBV's are the building blocks for contemporary web user analysis. Their web site behavioural information content can help predict consumer preferences and interests, which in turn improve web site structure and content. An advanced application is to generate online browsing navigation and content recommendations for the users, i.e., to personalize the user-site interactions which is essential for the new web-based systems.

Chapter 8

Acquiring and maintaining knowledge extracted from web data

*To acquire knowledge, one must study;
but to acquire wisdom, one must observe.*

Marilyn vos Savant

The last few years have witnessed an explosion in business conducted via the Web, illustrated by the growth on the number of web sites and visits to these sites. The result is a massive and growing quantity of web data.

Chapter 3 explained the value of the KDD techniques in the extraction of significant patterns from web data; that section focused on data mining and on the adaptation of data mining techniques to process web data. Many algorithms and systems have been proposed for analyzing the information that users leave behind after navigating through a particular web site.[155, 136, 176, 210, 245, 233].

These techniques provide user behavior patterns and preferences that will be validated by human experts, who can often suggest ways about how the patterns are to be used. One result is the development of web personalization systems, where the knowledge representation should be implemented easily and by using common programming languages, e.g., Perl, PHP, Java, etc. The knowledge representation

must consider changes in and to the web itself, i.e., changes in the web site structure and content, as well as the user behavior.

This chapter introduces a methodology to represent knowledge extracted from web data and its maintenance as a Knowledge Base (KB) structure [72], which consists of two parts - a Pattern Repository and a Rule Repository. The Rule Repository contains rules about how patterns are to be used. Finally, the chapter looks at the role the KB plays within an adaptive web site.

8.1 Knowledge Representation

The value and purpose of a web mining tool is to uncover significant patterns about the user behavior and preferences. However, the patterns need interpretation and a human expert is required for validation and to provide a rough indication about how the patterns should be used. This approach has pros and cons; for example, if the expert leaves the institution he will usually take the expertise with him.

Thus, it is important to find non-human or machine alternatives to complement human judgement in order to avoid such a situation. The first step is to organize what is known, using a formal language [76]; Knowledge Representation (KR) is the first step in developing an automatic system, using the primitive knowledge discovered at the web site. KR is not a trivial task. The underlying concepts can best be understood by examining its five dimensions [68], described below.

8.1.1 Fundamental roles of knowledge representation

Knowledge Representation is “how an entity sees the world”, understands a situation and prepares satisfactory action. In short, it is the ability to infer new perceptions from old ones.

The more complex a definition the more dimensions are in play. According to

[68], the five key dimensions of KR are:

Surrogate. A knowledge representation needs a surrogate about the elements that compose the external world. This allows an entity to determine the consequences of applying an action, i.e., to reason about it.

Set of ontological commitments. The choice of representation is to make a set of ontological commitments. All knowledge representations are approximations to reality. In this sense, KR is a method, with criteria such as “what we want to perceive” as a filter. For example, a general representation of voice production is to think in terms of a semi-periodic source (glottal excitation), vocal tract, lip radiation and the integration of these components. However, if the focus is on vocal track dynamics only, it can be explained as “the glottal excitation wave hits with the walls of the vocal track, generating a composed wave”, which can be understood as a set of ontological commitments.

Fragmentary theory of intelligent reasoning. Reasoning for the purpose of this activity has three basic components: the representation of the intelligent inference; the set of inferences to represent sanctions; and the set of recommendations implied by the inference. For example, in the classic economic theory, consumers make intelligent decisions based on the information about product characteristics and prices. When the price changes, so does the intention to purchase.

Medium for pragmatically efficient computation. As thinking machines are in essence computational processes, the challenge to efficient programming is to properly represent the world. Independent of the language used, it is necessary to correctly represent the problem and create a manageable and efficient code, i.e., not requiring redundant and high data processing capacities.

Medium of human expression. Knowledge is expressed through a medium such as the spoken language, written text, arts, etc. A communication interface is necessary if a human being is to interact with an intelligent system.

R_1 :	If VisitPage(p_1) and SpentTime(t_1) Then RecommendationPage(p_{10})
R_2 :	If BelongCluster(c_1) Then RecommendationPage(p_3, p_5, p_8)
R_3 :	If CountPageVisit(p_i) < D Then DeletPage(p_i)
...	...
R_n :	If ...

Figure 8.1: Knowledge representation using rules

8.1.2 Rules

Given the assumption of ontological commitments for knowledge representation, there must be rules as to how patterns are to be discovered and defined [44].

Rules express recommendations, directives and strategies. In computational terms, they are expressed as instructions in the form: **If** < condition > **then** < recommendation > . These instructions can be used to represent human expert knowledge. However, when the set of rules grow above a certain limit, it is difficult to decide which rule should be applied in a given situation.

The rules associate actions and recommendations by matching facts and conditions, as shown in Fig. 8.1.

In this example, if the user visits the page p_1 and spends t_1 time, then the recommendation is “go to the page p_{10} ”. If the user’s browsing behavior belongs to cluster c_1 , then the recommended pages to visit are p_3, p_5, p_8 .

Many programming languages can be used to implement rule based systems but differ in their ability to add and delete rules. Some have been developed for this specific purpose (e.g. Lisp and Prolog) and are less satisfactory with respect to facilities for implementing effective interchange protocols with applications.

8.1.3 Knowledge repository

What is the best way to maintain web related knowledge? The most common answer is to use a repository, similar to that used for data (patterns). However, this knowledge is not raw data but corresponds to patterns discovered after processing data and then translated into rules about how the patterns are to be used.

The Knowledge Base (KB) is a general structure for storing facts (patterns) and rules about how to use them. A typical representation will try to keep track of rules that share common perspectives [72].

In practice, the KB must be able to maintain rules in a straightforward way to ease implementation. This is complicated when, as in the case of web knowledge, the problem conditions change over time.

8.2 Representing and maintaining knowledge

The main idea behind a personalization system is to adapt web-based structure and content to cater to user choices. Thus, personalization requires information about user behavior before performing the adaptation. In web personalization, given a particular situation, the system proposes an action (recommendation). In others words, “*if* these facts happened”, “*then* this is the action to be performed”.

Web-based personalization system have be designed so that the site-user interaction is based on predefined actions, the result of the comparison of current to previous user behavior.

Previous behaviour is based on pattern extraction from web data supplemented by an expert who accepts or rejects the patterns. Isolated patterns have to be replaced by something more general, in order to predict a future action. However, as even general patterns could become too unweildy because of their size, care must be

taking in fixing the scope of the application. The stored knowledge must be used to give recommendations to the range of web users (e.g. occasional visitors, current customers, web masters) who have contact with the site.

These recommendations take the following pragmatic form:

- Offline. These cover changes in the web site when the web site is down. The modification are proposed to the web master or the person responsible for the web site.
- Online. These cover navigation hints for the current web site user. They are performed in real time during the user session.

Offline recommendations focus on the web site structure and content, i.e. the best hyperlink structure for the site and the most attractive content (free text, video, image, audio, colors, etc.). Online recommendations focus on future user navigation; once the user has visited a set of pages, the system will recommend other pages to be visited.

For either mode, browsing behaviour has the following characteristics:

- Different users have distinct goals;
- The behavior of a user changes over time;
- As site content expands, accumulating an increasing amount of pages and links, web pages need to be restructured according to new needs.

Recommendations using adaptive hypermedia [39] are created by understanding the preferences and experience of individual users, often by questionnaires, usability tests, and analyzing user-application interaction. A good tutorial is an example of adaptive hypermedia, as it incorporates past experience to adapt the current lesson for

a new student. In general, this kind of system uses a rule base engine for adaptation which can be fixed for the duration of the system; so that when a new version is released it is necessary to update. It is easier to update these rules in stable rather than dynamic environments and where the system needs regular modifications.

An ambitious example of adaptive hypermedia is that of web personalized systems [133]. This is a not an easy task because web based systems have high variability in structure and content, so the set of adaptation rules is likely to be large. Further, a recommendation could become obsolete quickly because of frequent changes to web site itself and the user's interests.

Some authors [72, 258] advocate the creation of KBs to store web data to meet this issue. In [238], the KB maintains the patterns, stored in a data base like repository; and the rules are an independent program, that interacts with the pattern repository to prepare navigation recommendations. As the repository contains patterns discovered at different times, it is convenient to use data warehouse architecture and generate the necessary parametric rules.

8.3 Knowledge web users

Web data users are either human beings or computer systems (inference engines).

Human users can be divided into web site users or maintainers. The former encompasses visitors, customers and those that follow system recommendations about navigation and content. The second group are web masters, web site designers, business managers and persons who provide support to modify web site structure and content.

Computer systems can personalize a user's navigation in a web site; so a Computerized Recommender system would use the knowledge extracted from web data to create the online navigation recommendations.

When web knowledge is contained in an alternative system, e.g. a KB, the human users can review the hyperlink structure and content recommendations for implementing offline changes in the web site. On the other hand, computer systems, like web-based personalized systems, obtain information through program language queries; the recommendations are then sent to the user.

8.4 A framework to maintain knowledge extracted from web data

This section introduces a practical example of the KDD process as a framework by which to analyze user browsing behavior and web site preferences.

The framework consists of two repositories to store the information and knowledge extracted from web data. Both structures provide offline and online recommendations to web users.

8.4.1 Overview

Fig. 8.2, shows a framework to acquire, maintain and use knowledge about the visitor behavior [238, 241, 240]. The idea is to use Webhouse architecture to create a Web Information Repository (WIR) for storing information to be analyzed; and a Knowledge Base (KB) which contains the results together with additional domain knowledge provided by experts. This allows a Computerized Recommender System (CRS) to re-organize the web site structure and content changes offline - and make online navigation recommendations. The WIR contains information which describes the behavior of the web site users and the content of each web page [32, 232]. The data sources are web log registers (dynamic data) and text contained in web pages (static information). The former is based on the structure given by W3C and the latter is the web site itself, i.e., a set of web pages with a logical structure. Following the preprocessing stage, the relevant information is loaded into the WIR.

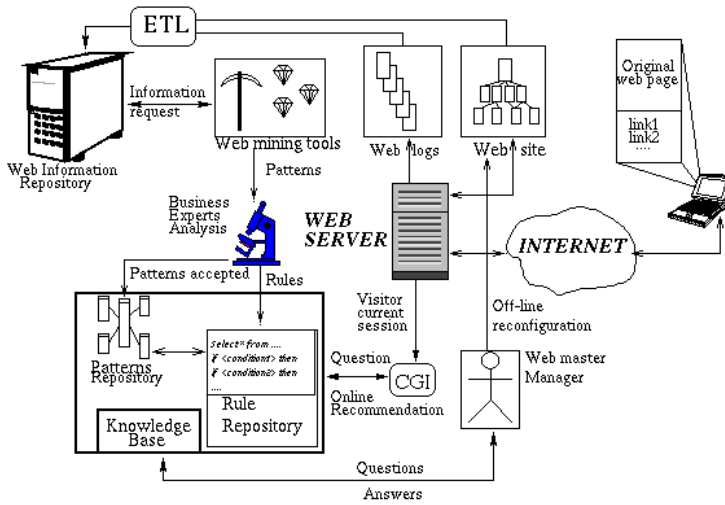


Figure 8.2: A framework to acquire and maintain knowledge extracted from web data

Due to high volume of data to be processed, a RDBMS [82] with high capacities to implement the WIR is recommended. Then, the Extraction, Transformation and Loading (ETL) process can be programmed, in part, using the database engine's language. It is also proposed that the process is facilitated by programming in a language that supports regular expressions for filtering web log registers. There are two good choices: **Perl** and **Php**, both having strong interfaces with any RDBMS.

By applying web mining techniques to the WIR, it is possible to find unknown and hidden patterns [230]. This task is performed using an algorithm that directly interacts with the WIR via the communication interface of the RDBMS.

The patterns extracted by the web mining tools should be validated by a domain expert and finally loaded onto the Pattern Repository. The specific usage of the patterns are implemented as rules which are loaded into the Rule Repository.

Pattern and rule repositories form the complete structure of the KB, which is used to recommend different kinds of web site modifications. As the repositories contain

past behaviour patterns, proposed changes can be checked against them.

One limitation is the possibility that users reject the changes as too invasive. Thus prior to confirming the changes to structure and content, the recommendations should be reviewed by an expert or webmaster [126].

The approach described here has two types of potential “knowledge users”: human beings or artificial systems. In the first case, the human users consult both repositories as a Decision Support System (DSS) and propose changes in the web site. These changes are usually performed by hand, although part of them can be automated. Artificial systems use mainly the pattern repository and return navigation recommendations as a set of links to web pages. Then, dynamic web pages will incorporate these information links sent to users.

The framework components are outlined below.

8.4.2 The Web Information Repository

Section 4.6 introduced the WIR’s generic structure based on webhouse architecture. This structure stored information about web site visitors, i.e., those users for which clickstream behavior is available only, without personal data, like name, sex, purchase behavior etc. In the case of user customers, additional tables were to be added to the WIR to store new data.

The star query model [128] was selected to store information generated by the transformation step in the WIR. As seen on Fig. 4.6, the WIR contains a table with three measures: the time spent on each page, the number of real sessions per page and the total bytes transmitted by the visitor session. These are additive measures and **OLAP** techniques [32, 33, 111] can be applied to obtain statistics about the visitor behavior. They are however only complements to web mining tools.

The interaction of the web mining tools with the WIR is performed as SQL

queries that return input for the web mining algorithms.

8.4.3 The Knowledge Base

Web mining techniques have the potential to reveal interesting patterns about the user behavior in a given web site, by analyzing the information stored in the WIR [231]. Experts in the particular application domains can interpret the patterns and suggest ways to improve the site structure and content as well as contribute to pattern extraction rules for online navigation recommendation modules.

The KB [44] contains data developed from the use of “if-then-else” rules based on the discovered patterns. Fig. 8.3 shows the general structure of the KB [237, 240]; composed by a Pattern Repository, which stores patterns discovered, and a Rule Repository, to store general rules about how to use the patterns.

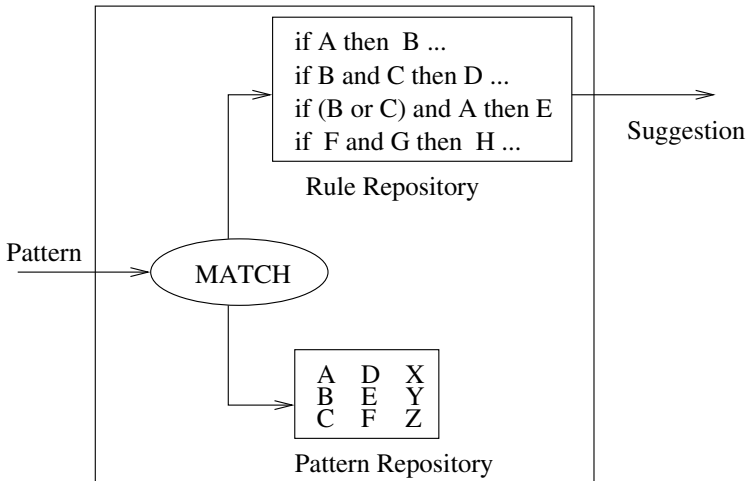


Figure 8.3: A generic Knowledge Base Structure

A search takes place when a pattern is presented to the KB, to find similar

patterns in the Patterns Repository. These patterns are used to select the set of rules that create the recommendations to be sent to users.

8.4.3.1 Pattern Repository

In the literature, it seems that web access data is that which is principally stored, see for example [32, 111]. We propose to store the discovered patterns as well.

The Pattern Repository stores the patterns revealed from the WIR by applying web mining techniques.

Fig. 8.4 shows a generic model of the Pattern Repository, which is based on the data mart architecture.

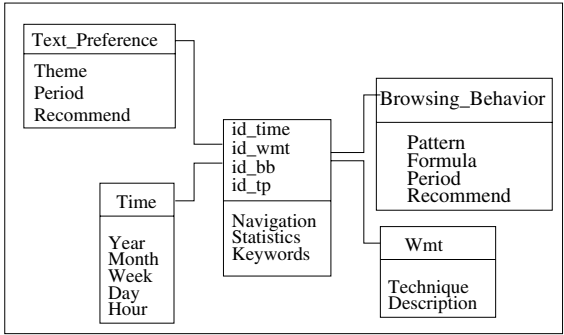


Figure 8.4: A generic Pattern Repository model

The pattern extraction process uses a matching function (column **formula** in table **Browsing_behavior**) to find best fit patterns, within the Pattern Repository, for the sample presented to the system. This repository can also be implemented using the data mart architecture of the star model. The fact table in Fig. 8.4 shows **navigation, statistics** and **keywords** measures. These measures are non-additives [128] and contain the web page navigation recommendations, related statistics, such

as the percentage of visits in the period under investigation and the web site keywords discovered. The dimensional table **Time** contains the date of application of the web mining technique over the WIR. The **browsing_behavior** table contains patterns about user browsing behavior. The *formula* column stores the specific expression used for feature vector comparisons, and the *description* column contains the details. The *period* column contains the time when the data to be analyzed was generated, e.g. “01-Jan-2003 to 31-Mar-2003” and in the *recommend* column suggests offline changes for helping users negotiate the web site structure. The **Text_preferences** table contains, in the *theme* column, a context description of the web site keywords. For instance, a context for keywords related to credit cards is “promotion about VISA credit card”. In the same table, the *recommend* column contains suggestions keyword inclusion in a textual fragment like a paragraph. The table **Wmt** (Web mining technique) stores the applied mining technique, e.g., “Self-Organizing Feature Map (SOFM) K-means, etc.”

After consulting the KB, the Pattern Repository returns a set of possible web pages to be suggested. Based on this set and additional information, such as statistics of accessed web pages, the Rule Repository makes the recommendations.

8.4.3.2 Rule Repository

The Rule Repository is used to recommend a page from the current web site, i.e. to make a navigation suggestion. The visited pages and time spent on them during a session can be found by using an online detection mechanism like a cookie.

With this data, the current visit is matched to previous stored visit patterns. This requires a minimum number of pages visited to understand the current user’s behavior.

Then, the Rule Repository is applied to the matching results to give an online navigation recommendation concerning the next page to be visited, based on the history of previous navigation patterns.

If the recommended page is not directly connected to the current page, but the suggestion is accepted by the user, then new knowledge about preferences can be generated. This can be used to reconfigure the Web site, by reorganizing the links among the pages.

```

select navigation, statistics, formula(pattern,n) into S
from pr_fact, time, browsing_behavior, wmt where "star join" and
"fix technique" and "fix time" and formula(pattern,n) >  $\epsilon$ ;
...
if S is empty then
    send("no suggestion");
...
while S not empty loop
    if S.navigation  $\notin$  actual_web_site then
        S.navigation = compare_page(ws,S.navigation);
    ...
    if S.navigation  $\neq$  last_page_visited and S.statics >  $\delta$  then
        send(S.navigation);
    ...
end loop

```

Figure 8.5: Sample of pseudo-code from Rule Repository

Fig. 8.5 shows a part of the rule set from Rule Repository. The parameter ϵ is used to identify those patterns that are "close enough" to the current visit. The parameter γ filters the recommendations whose statistics are above a given threshold, e.g. the page should have a minimum percentage of visits.

Since the Pattern Repository contains historical information, a recommended page may not appear in the current web site. In this case, the function "compare_page" determines the current web site page content with the recommended page.

8.5 Integration with adaptive web sites

Recall that this framework has two types of knowledge users: human beings and artificial systems, like inference engines. The human agent, by using SQL queries, reviews suggested structural and content proposals about older, current and newer versions of the web site and makes offline recommendations.

In the second case, there must be an interface between the web server and the KB to prepare an online navigation recommendation. Note that the reply uses HTML format, in order to avoid interpretation problems with the web browser.

By using Common Gateway Interface (CGI) specification, the web server can interact with an external system. When the external system is a database, there is an CGI interface - in the case of Perl programming language, the DBI/DBD interface allows communication between the database and the web server. In fact, the CGI is the practical realization of the Computerized Recommender System (CRS), used by the web site for personalizing the user session, as shown in Fig. 8.2. The CRS output contains the HTML navigation recommendations, which is the input of the web server and whose output is the web page to be set to the user.

Additionally, the KB and the WIR can be consulted by a human expert. In this case, reports from the WIR can be prepared using free-ware and commercial tools. For instance, Webtrends¹ prepares various statistics from web log sessions, and Oracle Discovery is a general tool for preparing reports from a relational database.

The queries to the KB can be performed as SQL instructions, or a particular interface can be developed. Offline changes to the web site structure and content are facilitated by using two repositories.

¹<http://www.netiq.com/webtrends/default.asp>

8.6 Summary

This chapter introduced a generic framework for acquiring and maintaining web data.

A Web Information Repository (WIR) and a Knowledge Base (KB) are key components. The former contains user sessions, the visited pages and statistics about the accessed pages and uses data mart architecture and the star model. The latter (KB) is a more complex structure that stores discovered patterns and rules about how to use them. Both elements are maintained separately in a Pattern Repository and a Rule Repository, respectively. As web site and user behavior vary greatly, the classical rule base representation (rigid rules) generates too many rules, complicating the rules storage. A set of parametric rules solve this problem and that follow the practice of consulting the Pattern Repository to prepare recommendations.

These can be classified as offline and online recommendations. Offline recommendations are performed manually and consist of changes in the web site structure and content. Here, the Pattern Repository and the WIR are consulted to create an appropriate recommendation.

Online recommendations are principally navigation recommendations to the user, i.e., links to pages to be visited. These are provided by an automatic computerized recommender system, that interacts with the KB and returns an HTML output file with the navigation recommendations. This is the input file to the web server that sends the pages to the users. Here, there is a big challenge - the user may not agree with the recommendation, which might brake the web site structure (e.g. show a deeper page) and generate navigation problems for the visitor.

Once the online navigation recommendations are accepted or rejected by users, this new knowledge can be used for web site reconfiguration, i.e., to reorganize hyperlinks among web pages. Since the WIR and the KB store past data, it is possible to check, in principle, if an expert's previous recommendations were successful or not.

Chapter 9

A framework for developing adaptive web sites

*It is not the strongest of the species that survives,
nor the most intelligent, but the one most responsive to change.*

Charles Darwin

This chapter introduces a general framework about how to use the algorithms, methods and approaches reviewed and developed in this book for the construction of an adaptive web site. We explain the complete process with an example, starting from the selection of web data sources and going up to the *a priori* tests that measure the operational effectiveness of the adaptive web site with real users.

Any analysis of the web site user behaviour must begin by collecting original data from the real web site, as it is not possible to simulate a user session in the web. There may be a problem here as many interesting sites for web mining purposes, with a rich hyperlink structure and content, are owned by commercial companies. They are unlikely to release confidential and strategic data, even for academic proposes. This problem can sometimes be resolved by writing a non-disclosure agreement with the proprietary company regarding the specific bank case presented in this chapter, which then allowed the use of data originated from the bank web site for experimental pur-

poses. Examples about using their data for pattern extraction process, by employing web mining algorithms, and the construction of an adaptive web site are presented.

9.1 The adaptive web site proposal

Figure 9.1 shows a framework for acquiring, maintaining and managing knowledge about web-user behavior. On the left side there are three repositories: the Web Information Repository (WIR), the Pattern Repository (PR) and the Rule Repository (RR). The WIR stores the web data to be analyzed while the PR stores the analysis results, and the RR contains domain knowledge drawn from human experts. The two final structures make up the Knowledge Base (KB) about user behavior. This framework allows making online navigation recommendations, as well as offline changes to the web site structure and the text contents.

The WIR can be implemented under the data mart architecture by applying the star model. It contains information from web data, for example, user session information (visited pages, time spent, page sequence, etc.) and the web page contents. By construction, the repository stores historical information and allows the direct application of web mining tools at any time. By applying web mining techniques to WIR, it is possible to discover new and hidden knowledge about user browsing behavior and preferences [234].

As a first step, the behavior patterns extracted by the web mining tools should be validated by a business expert prior to being loaded onto the PR. Then the behavior patterns are converted into rules and loaded into RR. Both PR and RR constitute the KB's complete structure [44], and are then used to make recommendations. Both repositories hold historical information, so that the impact of future web sites changes can be measured against past changes and used to extrapolate future behaviour patterns.

This procedure allows for two different users - human beings and artificial sys-

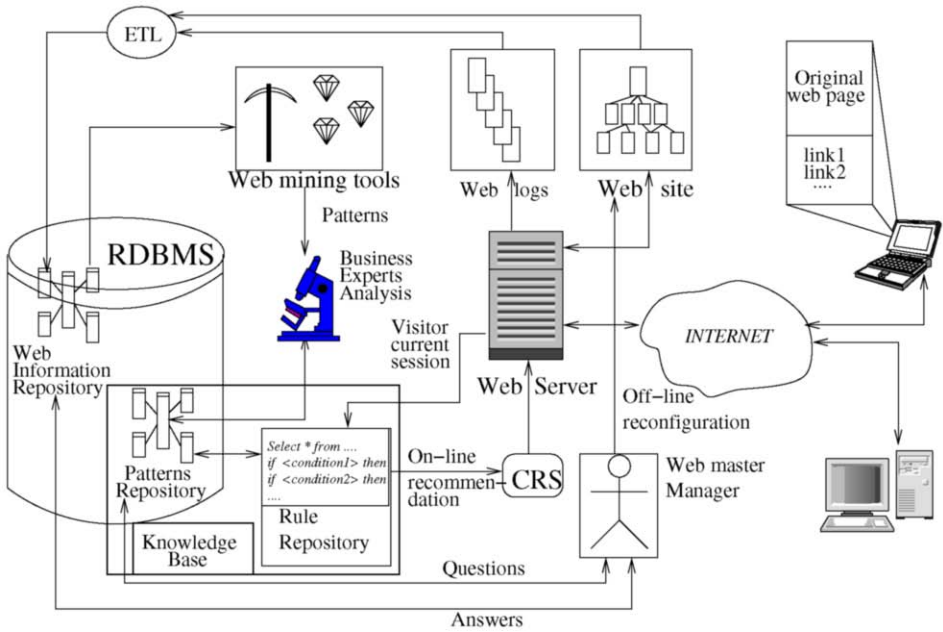


Figure 9.1: A framework for constructing adaptive web sites

tems. Human beings consult the KB as a Decision Support System (DSS) and propose changes to and in the web site. These are usually made manually, although some of them can be automated. Artificial systems use the PR and return navigation recommendations as a set of links to web pages. In figure 9.1, the Computerized Recommender System (CRS) creates a dynamic web page containing the online navigation recommendations, received as input by the web server and then sent back to the users.

9.2 Selecting web data

For experimental purposes, the selected web site should be complex with respect to several features: number of visits, periodic updating (preferably monthly in order

to study the user reaction to changes) and rich in text content. The web page of a Chilean virtual bank (no physical branches, all transactions undertaken electronically) met these criteria. As noted, the authors signed a non-disclosure agreement with the bank and they are not in a position to provide its name.

The main characteristics of the bank web site are the following; presented in Spanish, with 217 static web pages and approximately eight million raw web log registers for the period under consideration, January-March 2003. The bank web site was designed for two types of users - visitors and customers. A visitor is defined as an anonymous user, normally looking for information about credits, investments, how to open an account, etc. For the visitor, the bank web site consists of information pages which can be accessed by any user. A bank customer has access to a special and hidden part of the web site, via https, an encrypted version of the http.

Figure 9.2 shows the home page. On the top right side corner there is a login for the web site customer to enter. When a customer is logged on, a security http protocol is activated and a cookie mechanism is used to identify the customer and record his or her session.

At the bottom right side corners of the home page, more information is provided about the institution and its products. When the user clicks on an option, a page appears, as shown in figure 9.2. All user options are intended to persuade the visitor to become a bank customer. Different bank products are listed on the right side of figure 9.2, and promotions placed at the center. The layout of the bank web site is shown in figure 9.3. As can be seen, there are four levels, beginning with the “home page”; at the second level, there are three main options, including the “customer zone”, i.e., the area where the customer is validated.

The confidentiality agreement does not allow us to show the structure of the customer zone. The other two options are accessible to visitors. The third levels shows information about products and services and the fourth contains specific product information.

The behavior of a user at the bank web site is analysed in two ways. First, by using web log files which contain data about visitor and customer browsing behavior. This data requires prior reconstruction and cleaning before web mining tools are applied. Second, web data is the web site itself, specifically the web page text content - this also needs preprocessing and cleaning.

Before applying any task, it is helpful to recall the dynamic nature of a web site. Often when web page content is changed, the page's physical web site name is



Figure 9.2: Home page of the virtual bank

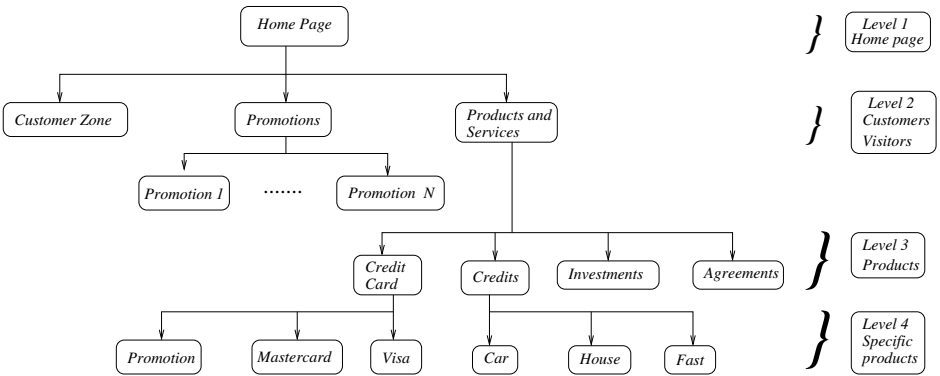


Figure 9.3: Bank’s web site layout

kept unchanged. For instance, the “index.html page may have had ten changes in the last month but continues being called “index.html”. This situation can complicate tracking page content changes but it is important for understanding user information preferences. This problem can be avoided by maintaining the older page versions and recording the date when any change took place.

9.3 Extracting information from web data

A consolidated and historical WIR is created by a specific data mart architecture for storing the information related to user behaviour in the bank web site. This architecture is based on a star model and contains the historical data from web site visits. Its fact table contains various additive measures to facilitate the intended web mining tasks, whereas the dimension tables store the necessary parameters that allow the analysis (e.g. period of analysis, range of pages within a session, etc.).

The repository stores different kinds of information derived from web site visits such as the time spent on each page and the sequence of pages followed per session. As the WIR has a flexible structure, other parameters that describe user navigation behaviour can be added so that it becomes a flexible research platform for various

kinds of analyses.

9.3.1 The star model used for the creation of the WIR

The WIR contains the main structure for loading the web data information, after cleaning and preprocessing. The star model was selected because of the large amount of information to be stored and implemented in a RDBMS engine; this can be either a commercial or freeware tool, depending on the queries requirements. For effective online recommendations the query engine that retrieves information from the WIR cannot be too slow. It may be necessary to improve the query engines performance by using acceleration objects like index and materialized views, which are usually incorporated only in commercial tools.

Figure 9.4 illustrates the WIR table's composition. Three measures are stored in the fact table; the time spent on each page, and identifier number which relates the user session with the page visited and the total number of bytes transmitted when the page was requested.

The star model is composed of seven dimensional tables. **Time** and **Calendar** contain the time stamp and date, respectively, and **Pages** contains the URL and the text of each page. The table **Pagedistance** efficiently stores the distances between web pages. The **Session** table shows the IP and Agent that identify a real session, the table **Referrer** contains the object that refers to the page and, finally, the table **Method** contains the access method used.

9.3.2 Session reconstruction process

Fig 9.5 shows part of the bank's web log registers and includes both identified customers and anonymous visitors. Customers access the site through a security connection, using a SSL protocol that allows the storage of an identification value in the authuser parameter in the web log file. Another way of identifying users is by cookies,

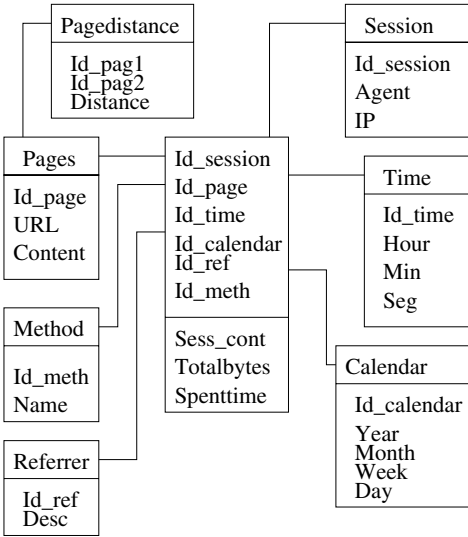


Figure 9.4: Data Mart for web information using the star model

but sometimes these are deactivated by users in their browsers. In this case it will be necessary to reconstruct the visitor session.

During the session reconstruction process, filters are applied to the web logs registers. In this particular case, only the registers requesting web pages are used to analyze the site specific user behavior. It is also important to clean abnormal sessions, for example web crawlers, as is shown in Fig. 9.5 line 4, where a robot that belongs to Google is detected.

#	IP	Id	A	Time	Method/URL/Protocol	Statu	Byte	Referer	Agent
1	184.77.129.50	-	-	12/Apr/2003:23:47:44	GET /img/tab.gif HTTP/1.1	200	89	http://www.thebank.cl	MSIE 6.0; Windows 98
2	200.28.206.200	-	20	12/Apr/2003:23:48:31	GET transa/info.htm HTTP/1.1	200	144	/infoeco/info.html	MSIE 4.01; Windows 95
3	200.86.248.170	-	-	12/Apr/2003:23:48:37	GET /img/gen.gif HTTP/1.1	304	0	/ofer/wines/	MSIE 6.0; Windows 98
4	68.249.65.97	-	-	12/Apr/2003:23:48:41	GET /index.htm HTTP/1.1	200	88	-	Googlebot/2.1; google.com/bot.html
5	216.241.8.179	-	31	12/Apr/2003:23:50:03	GET /b/infoeco/card.htm HTTP/1.1	200	210	/b/infoeco/prom/	MSIE 6.0; Windows NT 5.1
6	184.77.129.50	-	-	12/Apr/2003:23:48:34	GET /b/infoeco/ HTTP/1.1	200	186	/b/infoeco/card.htm	MSIE 6.0; Windows 98
7	200.28.206.200	-	20	12/Apr/2003:23:51:13	GET transa/account.htm HTTP/1.1	200	180	/transa/info.htm	MSIE 4.01; Windows 95
8	216.241.8.179	-	31	12/Apr/2003:23:51:23	GET /b/infoeco/ind.htm HTTP/1.1	200	300	/b/infoeco/card.htm	MSIE 6.0; Windows NT 5.1
9	200.86.248.170	-	-	12/Apr/2003:23:51:41	GET /prom/wine.html HTTP/1.1	404	0	/ofer/wines/	MSIE 6.0; Windows 98
10	184.77.129.50	-	44	12/Apr/2003:23:52:04	GET /b/infoeco/ind.htm HTTP/1.1	200	186	/b/infoeco/	MSIE 6.0; Windows 98

Figure 9.5: A raw web log file from the bank web site

The raw web logs data covers four months of transactions, with approximately eight millions registers. Only registers related to web pages are considered for session reconstruction and user behavior analysis purposes; information that points to other objects, like pictures, sounds, etc., will be cleaned.

In this case study example, we are interested in the analysis of the web user, either customer or visitor. The customer's identification information is used for session reconstruction purposes only, and is obtained directly from the web log registers. Anonymous visitor behaviour is more complex and so the sessionization process introduced in section 2.2.1 is applied. This process can be implemented in any language, but it is desirable to use those languages that support common expressions and predefined organizer or acceleration objects (e.g. indexes, arrays, etc.) to allow for sorting and grouping tasks. The RDBMS where the WIR is implemented has these characteristics and, additionally, a Data Staging Area (DSA) [32, 128] defined as tables, to process the web log data. It is composed of two tables: **Weblogs** and **Logclean** (see figure 9.6).

Weblogs		Logclean	
Ip	varchar2(18)	Ip	varchar2(18)
TimeStamp	date	TimeStamp	date
Method	varchar2(20)	Bytes	number(8)
Status	number(4)	Url	number(4)
Bytes	number(8)	Agent	number(2)
Url	varchar2(20)	Session	number(4)
Agent	varchar2(20)	Timespent	number(4)

Figure 9.6: Data staging area for session reconstruction process

To reconstruct a visitor's session, first the Weblog table is loaded and which, by construction, receives the corresponding parameters in each of the web logs files in its columns - specifically the IP address and agent, time stamp, embedded session Ids, method, status, software agents, bytes transmitted, and objects required (pages,

pictures, movies, etc). Only the registers without an error code and those with an URL parameter link to web page objects are considered. Other objects, such as pictures, sound, and links are discarded.

The loading task can be performed in the same way, as the majority of the RDBMs have a utility program that transfers the registers from a file to tables in the database. To do so, a prior operation is required, where the web logs are transformed into a standard file for the utility program, usually a collection of parameters separated by semicolon. Figure 9.3.2 shows a regular expression used for splitting a web log register into elemental components. This data is assigned to local variables which can be used to create a file with a desired format. The web log cleaning task, noted above, is undertaken by using the code's local variables,

```
my ($host, $ident_user, $auth_user, $date, $time, $time_zone,
    $method, $url, $protocol, $status, $bytes, $ref, $agent, $query)=
/^(\\S+) (\\S+) (\\S+) \\[([[::]]+):(\\d+:\\d+:\\d+) ([^\\]]+))\\ " (\\S+)
(\\.+?) (\\S+)" (\\S+ ) (\\S+) "(\\S+)" "(\\.+?)" (\\S+)$/;

% Cleaning the web log registers

%Creation of the file with the desire format
print "$host;$date;$time;$url;$protocol;$status;$ref;$agent \\n"
```

Figure 9.7: A regular expression for processing web logs

The session reconstruction process uses a time oriented heuristic [63] that considers 30 minutes to be the session's maximum duration. By using a pseudo-code such as that shown in figure 9.8, the registers are grouped by IP and Agent, sorted by date and stored in a data structure known as "cursor". It is similar to the stream load in a RAM memory. This code only considers those sessions with a minimum number of registers (SQL instruction, line 3). So a session is not considered to be useful if it has less than three registers or if the session contains a greater number of page visits than the allowed maximum as in a compound session, when several real users arrive

at the web site but use the same IP address (possibly the firewall IP) and agent. The session reconstruction identification problem is discussed more fully in section 2.2)

At this point in the session reconstruction process, it is convenient to improve the sessionization algorithm by analyzing the session reconstructed. For example, it will be helpful to detect abnormal sessions, where a user appears to visit an enormous number of pages in a short time, as happens when a crawler system is retrieving pages for an indexing process. While some crawlers signal they are not real users in the agent parameter, this does not always occur.

The statistical analysis of the reconstructed sessions always contributes to the improvement of the final result. However, the unreal sessions can only be reduced statistically as there is no identification mechanism when working with visitor sessions.

Because is not possible to determine the time spent in the last pages visited by the user, one approach is to consider the average time spent on the other pages of the same session (lines 11 to 15 in Fig. 9.8).

The **Logclean** table contains the log registers grouped by IP and Agent together with a correlative number that symbolizes the identified session that is stored in the **session** column. From the identified sessions, it can be directly inferred which web site objects are visited by the users and in particular which web pages. About 16% of the users visited 10 or more pages and 18% less than 3 pages. In both cases, the sessions are discarded.

Customers usually visit three or more pages to access relevant information, such as their account balance. However, sometimes the customers may visit two pages only, which is uninteresting for analysis, and so these sessions also will be discarded.

```

1  cursor log is select ip, timestamp, method, url, protocol, status,bytes, ref, agent
2  from weblogs where status between 100 and 304
3  having count(*) > 3
4  order by ip,agent, timestamp;
5  ....
6  while log%found loop
7      difftime:=(new.timestamp - old.timestamp)*86400;
8      if (difftime < 5) then fetch log into new;
9      elsif ((old.ip = new.ip) and (old.agent = new.agent) and (difftime < 1800)
10     and (trans < vlarge)) then
11         insert into Logclean values (old.ip,old.timestamp, old.url, difftime,session);
12         totaltime:=totaltime+difftime;
13         numinsert:=numinsert+1;
14         trans:=trans+1;
15     elsif (totaltime !=0) then
16         insert into Logclean values (old.ip,old.timestamp, old.url,
17         totaltime/num insert, session);
18         session:=session+1;
19         totaltime:=0;
20         trans:=0;
21         numinsert:=1;
22     end if;
23 end loop;

```

Figure 9.8: SQL pseudocode for sessionization

9.3.3 Web page content preprocessing

By applying web page text filters, it was found that the complete web site contains $R=4,096$ different words to be used in the analysis. Regarding the word weights and the special words specification, the procedure introduced in section 7.2.1 was used, in order to calculate sw_i in equation 7.2. The data sources were:

1. The e-mails received in the call center platform. During the period under analysis, 2128 e-mails were received, containing a total of 8125 words following preprocessing and cleaning. From this set of words, only 1237 are to be found in the web site text content.

2. Marked words. Inside the web pages, 743 different words were found after applying the preprocessing and cleaning step.
3. Related web sites. Four web sites were considered, each of them with approximately 300 pages. The total number of different words was 9253, with 1842 of them contained in the web site text content.

After identifying the special words and their respective weights, it is possible to calculate the final weight for each word in the entire web site, by applying the Eq. 7.2. Then, the vector representation for all the pages in the site is obtained. By using the Eq. 7.3, the distance between all pairs of pages of the site can be calculated and stored in a 3-dimensional matrix. Figure 9.9 shows a plot of the upper part of the 3D matrix, since this matrix is triangular. A few clusters of the page contents can be clearly seen. This shows that the free text used in the construction of the web site is related.

Page distance

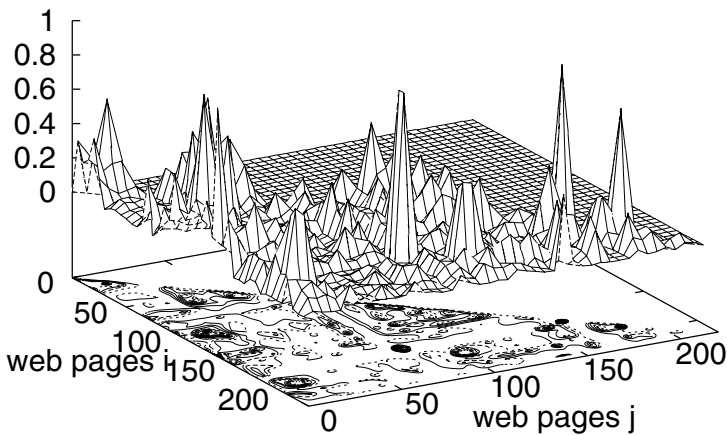


Figure 9.9: Similarity measure among web pages

9.4 Applying web mining techniques

Simply, the purpose of web mining algorithms is to extract patterns about the user navigation behavior and text content preferences. So a prior operation is required that creates user behavior vectors as an input into the algorithms. The vectors are created from user sessions, which are reconstructed by using data contained in the web logs. The length of the session, in terms of pages visited, is a key variable for user browser behaviour. To set the cardinality of the user behavior vector, heuristics are applied by using various statistics about the user session.

9.4.1 Analyzing the user browsing behavior

A simple way to perform a preliminary user browsing behavior analysis is by extracting data about the times the web site has been visited by users, for example the most visited pages, the user session length, etc. This information helps in the creation of the user behavior vector, to be used as an input for the web mining algorithms at the pattern extraction stage.

Section 5.3.2, introduced the potential of using clustering methods for behavior analysis. Two clustering algorithms were proposed: a SOFM with toroidal topology and a basic K-means. Both have in common the similarity measure used to compare user sessions, introduced in Eq. 7.8.

9.4.1.1 Applying statistics

From the statistics in Table 9.1, the typical session consists of five pages and ten minutes spent time on average. This data considers both customers and visitors. In the bank, 90% of all transactions are originated by customers.

Table 9.2 contains information about bank web page services and the average time spent in seconds by the user. The table does not refer to customer services

Table 9.1: Summary of the statistics of the bank web site

Indicator	Measure
Average number of pages visited per session	5
Average time spent per page in seconds	139
Maximum time spent per page in seconds	1800
Minimum time spent per page in seconds	3
Average time spent per session in seconds	445
Number of user behavior vectors	295,678

because this information is considered confidential by the bank. However, it can be noted that the customer services, in general, take more time than those shown in the table.

Table 9.2: The top ten pages ranked by average time spent per page

Page	Average time spent in seconds
Services: credits page	214
Services: online credit simulation	203
Services: cash advance options	198
Promotions: Investments	196
Services: multi credit simulation	186
Promotions: special discount	185
Credit card: special promotion	166
Services: house credit simulation	160
Remote services	159
Investments	134

Table 9.3 shows the most visited pages, excluding information about customer pages. The home page is common to both types of users, while the other pages are relative to the type of user. In the customer zone, similar options can be found.

9.4.1.2 Using SOFM for extracting navigation patterns

As noted, five pages were visited an average user session. As sessions with three or fewer pages visited and those with ten or more pages visited are filtered, we can set

Table 9.3: The ten most visited pages

Page	Amount of visits
Home page	530302
Services: main page	48892
Services: credits page	7519
Services: house credit	5697
Services: family plan	4143
Services: online help	3983
Services: consumer credit	3779
Agreements with other institutions	3355
Promotion: special credit	2435
Credit card	2301

six as the maximum number of components per user behavior vector. Vectors with 4 or 5 pages are filled up with blank pages up to the 6th component. The time spent at these blank pages was set to zero seconds. After applying these filters, approximately 200,000 vectors were obtained.

To facilitate this analysis, the pages in the web site were labelled with a number. Table 9.4 shows the main content of each page.

Table 9.4: Bank web site pages and their content

Pages	Content
1	Home page
2, ..., 65	Products and Services
66, ..., 98	Agreements with other institutions
99, ..., 115	Remote services
116, ..., 130	Credit cards
131, ..., 155	Promotions
156, ..., 184	Investments
185, ..., 217	Different kinds of credits

The SOFM used has 6 input neurons and 32*32 output neurons in the feature map, with a toroidal topology. The cluster identification is performed by using a visualization tool supported by a density cluster matrix, called the “*winner matrix*”.

It contains the number of times that the output neurons win, during the training of the SOFM. Using this information, the clusters found by the SOFM are shown in figure 9.10. The x and y axes represent the neuron's position, and the z axis is the neuron's winner frequency, with the scale adapted.

By checking the information contained in the winner matrix and Figure 9.10, eight clusters were identified; however only four of them were accepted by the business expert. The accept/reject criterion was based on whether the page sequence in the cluster centroid was really a feasible sequence, following the current web site hyperlink structure. If affirmative, the cluster was accepted. But if a hyperlink between two consecutive pages in the centroid does not exist, then the pages are substituted by the closest or nearly similar pages in the web site in terms of content. If the situation persists - that is there is no hyperlink between pages - then the cluster is definitively rejected.

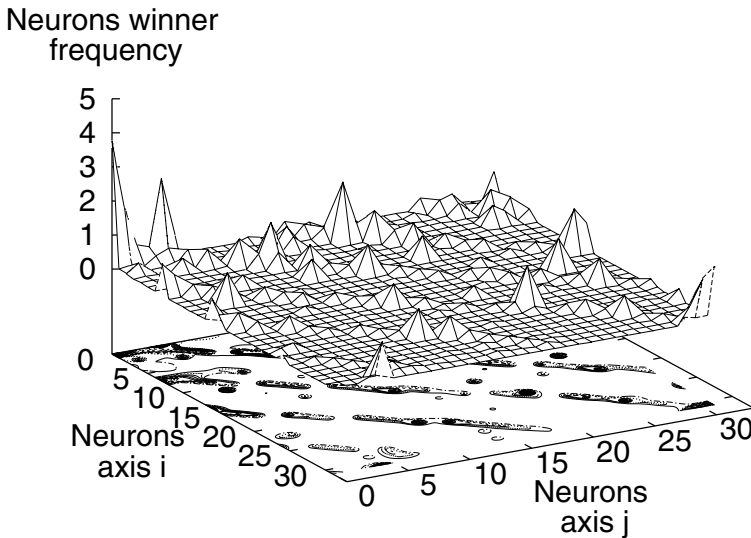


Figure 9.10: Clusters among user behavior vectors

Table 9.5 shows the clusters extracted by the SOFM and validated by the above

criterion. The second column contains the centroid of the cluster, represented by the sequence of visited pages, and the third column indicates the time spent in each centroid.

Table 9.5: User behavior clusters

Cluster	Visited Pages	Time spent in seconds
1	(1,3,8,9,147,190)	(40,67,175,113,184,43)
2	(100,101,126,128,30,58)	(20,69,40,63,107,10)
3	(70,86,150,186,137,97)	(4,61,35,5,65,97)
4	(157,169,180,101,105,1)	(5,80,121,108,30,5)

A simple cluster analysis shows the following results:

- Cluster 1. Users that are interested in general products and services offered by the bank.
- Cluster 2. Users search for information about credit cards.
- Cluster 3. Users are interested in agreements between the bank and other institutions.
- Cluster 4. Users are interested in investments and remote (distance) services offered by the bank

The patterns discovered in the cluster analysis form the basis for the online and offline recommendations, which are validated by a business expert. Note here the value of using simple statistics to support theories about web user behavior.

9.4.1.3 Using K-means for extracting navigation patterns

The principal idea here is to assign each feature vector to a set of given cluster centroids and then update the centroids. This procedure is repeated iteratively until a given stopping criterion is fulfilled (see section 3.5).

The number of clusters to be found (k) is a required input value for k-means. Since we found four real clusters with the SOFM, we used $k = 4$ as input value for k-means. We took a random selection from the original set of training vectors as the initial centroids.

By applying the k-means over the same set of user behavior vectors used in the training of the SOFM, the algorithm converged to the result shown in table 9.6.

Table 9.6: K-means user behavior clusters

Cluster	Visited Pages	Time spent in seconds
1	(2,29,45,112,120,154)	(20,69,35,126,134,90)
2	(200,135,10,50,132,128)	(3,101,108,130,20,13)
3	(1,100,114,128,141,148)	(4,76,35,8,89,107)
4	(131,135,156,182,118,7)	(25,62,134,103,154,43)

While cluster centroids in table 9.6 are different from those in table 9.5, a more detailed analysis of the assigned user behavior vectors showed similarities between the clusters found. However it cannot be concluded which method gives a better solution; both results are accepted and then interpreted by the business expert.

The performance of k-means depends directly on the similarity measure used. In this case it is complex and expensive to calculate when compared to the traditional Euclidean distance typically used in k-means.

9.4.2 Analyzing user text preferences

The “web site keywords” concept was introduced in section 7.4, as *a word or a set of words that makes the web page more attractive for the user*. The section discussed methods for identifying web site keywords, assuming a correlation between keywords and the maximum time spent per page/sessions, and introduced the concept of the Important Page Vector.

We fixed at 3 the vector's maximum dimension. Then, a SOFM with 3 input neurons and 32 output neurons was used to find clusters of Important Page Vectors.

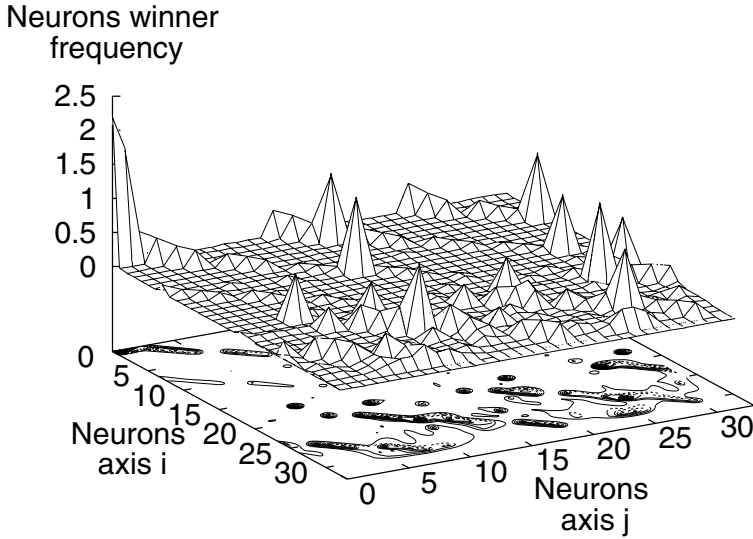


Figure 9.11: Clusters of important page vectors

Figure 9.11 shows the neurons positions within the SOFM on the x, y axes . The z axis is the normalized winning frequency of a neuron during training.

Figure 9.11, shows 12 main clusters which contain the information about the most important web site pages. However, only 8 were accepted by the business expert. The accept/reject criterion is simple; if the pages in the cluster centroid have the same main theme, then the cluster is accepted - otherwise it is rejected.

The cluster centroids are shown in table 9.7. The second column contains the center neurons(winner neuron) of each cluster and represents the most important pages visited.

A final step is required to get the web site keywords; to analyze which words in

Table 9.7: Important page vectors clusters

Cluster	Pages Visited
1	(6,8,190)
2	(100,128,30)
3	(86,150,97)
4	(101,105,1)
5	(3,9,147)
6	(100,126,58)
7	(70,186,137)
8	(157,169,180)

each cluster has a greater relative importance in the complete web site.

The keywords and their relative importance in each cluster are obtained by applying the equation 7.12. For example, if the cluster is $\zeta = \{6, 8, 190\}$, then $kw[i] = \sqrt[3]{m_{i6}m_{i8}m_{i190}}$, with $i = 1, \dots, R$.

Finally, by sorting the kw in descending order, we can select the k most important words for each cluster, for example $k = 8$.

We are not able to show the specific keywords because of the confidentiality agreement with the bank. For this reason the words are numbered. Table 9.8 shows the keywords found by the proposed method.

Cluster	Keywords	kw sorted by weight
1	$(w_8, w_{1254}, w_{64}, w_{878}, w_{238}, w_{126}, w_{3338}, w_{2343})$	$(2.51, 2.12, 1.41, 1.22, 0.98, 0.95, 0.9, 0.84)$
2	$(w_{200}, w_{2321}, w_{206}, w_{205}, w_{2757}, w_{3948}, w_{1746}, w_{1949})$	$(2.33, 2.22, 1.12, 1.01, 0.93, 0.91, 0.90, 0.89)$
3	$(w_{501}, w_{733}, w_{385}, w_{684}, w_{885}, w_{2326}, w_{3434}, w_{1564})$	$(2.84, 2.32, 2.14, 1.85, 1.58, 1.01, 0.92, 0.84)$
4	$(w_{3005}, w_{2048}, w_{505}, w_{3675}, w_{3545}, w_{2556}, w_{2543}, w_{2654})$	$(2.72, 2.12, 1.85, 1.52, 1.31, 0.95, 0.84, 0.74)$
5	$(w_{4003}, w_{449}, w_{895}, w_{867}, w_{2567}, w_{2456}, w_{767}, w_{458})$	$(2.54, 2.14, 1.98, 1.58, 1.38, 1.03, 0.91, 0.83)$
6	$(w_{105}, w_{959}, w_{212}, w_{2345}, w_{3456}, w_{3267}, w_{1876}, w_{384})$	$(2.64, 2.23, 1.84, 1.34, 1.11, 0.97, 0.89, 0.81)$
7	$(w_{345}, w_{156}, w_{387}, w_{387}, w_{458}, w_{789}, w_{1003}, w_{376})$	$(2.12, 1.87, 1.42, 1.13, 0.95, 0.87, 0.84, 0.78)$
8	$(w_{2323}, w_{1233}, w_{287}, w_{4087}, w_{594}, w_{587}, w_{2575}, w_{257})$	$(2.35, 1.93, 1.56, 1.32, 1.03, 0.92, 0.83, 0.76)$

Table 9.8: The 8 most important words per cluster

Table 9.9 shows a selected group of keywords from all clusters. Keywords on their own, however, do not make much sense. They need a web page context where they

could be used as special words, e.g. marked words to emphasize a concept or as link words to other pages.

Table 9.9: A part of the discovered keywords

#	Keywords	
1	Crédito	Credit
2	Hipotecario	House credit
3	Tarjeta	Credit Card
4	Promoción	Promotion
5	Concurso	Contest
6	Puntos	Points
7	Descuento	Discounts
8	Cuenta	Account

The specific recommendation is to use the keywords as “words to write” in a web page, i.e., the paragraphs written in the page should include some keywords and some could be linked to other pages.

Furthermore it is possible on the basis of this exercise to make recommendations about the text content. However, to reiterate, keywords do not work separately for they need a context. Reviewing Table 9.8, for each cluster, the discovered keyword could be used to rewrite a paragraph or an entire page. In addition, it is important to insert keywords to highlight specific concepts.

Keywords can also be used as index words for a search engine,i.e., some could be used to customize the crawler that visits web sites and load pages. Then, when a user is looking for a specific page in a search engine, the probability of getting the web site increases.

9.5 Using the extracted knowledge for creating recommendations

The application of web mining techniques to the bank's web data, or more accurately clustering algorithms, allows clusters to be identified which represent web site specific user behavior. The discovered clusters do however need human expertise to assess cluster quality and validate or reject the discovered clusters. In this bank example, they are the marketing manager, commercial manager, web masters and others with appropriate expertise.

The extracted patterns provide the platform for making recommendations for personalizing user experience at the web site and can be grouped into offline and online recommendations. This section explores examples of recommendations and ways of testing their effectiveness. Finally, the analysis of user behavior will be stored as patterns and rules (about how to use the patterns), which configure the extracted web behaviour information to create recommendations.

9.5.1 Offline recommendations

Offline recommendations propose hyperlinks to be added and/or eliminated from the current site; and the words to be used in the current and future pages as content recommendations. A few of the recommendations can be shown here and are currently being evaluated by the bank prior to their final implementation on the web site.

9.5.1.1 Structure recommendations

The user behavior clusters show the probabilities of visiting the bank product or service pages. The page position at the bank web site (second level in the web site layout, see figure 9.3) allows easy access to information. Hence this cluster must be kept.

The second cluster shown in table 9.5, corresponds to users with a significant interest in credit cards. The associated data confirm this interest, as the credit card product page is one of the ten most visited. This situation could be related to the Chilean economic situation (slight recession) at the time of this analysis and lower interest rates.

Reviewing the position of the web site credit card pages (third level), we can deduce that it is advisable to make a direct link from the home page.

The third cluster shown in table 9.5 corresponds to users interested in agreements between the bank and other institutions. However, determining which agreement is more attractive is difficult and the related statistics do not help to clarify this situation. A more detailed analysis is needed in this case.

The last cluster showed in table 9.5 corresponds to users interested in the investments offered by the bank. Here, if information about discounts for bank customers was added, it could act as a real incentive to transform users into customers.

Based on the clustering of similar visits, we proposed changes to the web site regarding the reconfiguration of the links structure. Some of these recommendations were:

Add links intra clusters. The idea is to improve page accessibility within each cluster from other pages belonging to the same cluster.

E.g.: Add a direct link from page 137 to page 150. Explanation - in cluster 3 many users spent considerable time on page 150 (35 seconds), then looked for a few seconds on the following page (186, only 5 seconds) and then stayed longer on page 137 (65 seconds).

Add links inter clusters. The idea is to improve the accessibility of pages belonging to different clusters that share many users in common.

E.g.: Add a direct link from page 105 to page 126. Explanation - in cluster 4

many users are interested in the pages of cluster 3. Users from cluster 2 that are interested in pages of cluster 3 also visit the page 126 of cluster 4.

Eliminate links. Inter cluster links that are rarely used can be eliminated.

E.g.: Eliminate the link from page 150 to page 186. This link caused “confusion” to many users in cluster 3.

For the bank it is vital for users to be able to find what they are looking for in the web site. These structure recommendations are to facilitate user information search, minimizing navigations and providing a shorter paths to specific contents.

9.5.1.2 Content recommendations

Web site keywords are concepts to motivate the users’ interests and make them visit the web site. They are to be judged within their context for as isolated words they may make little sense, since the clusters represents different contexts. The specific recommendation is to use the keywords as “words to write” in a web page.

Web site keywords can also be used as search engine index words, i.e., some of which could be used to customize crawlers that visit web sites and load pages. Then, when a user is looking for a specific page in the search engine, the probability of getting the web site increases.

As each page contains a specific text content, it is possible to associate the web site keyword to the page content; and from this suggest new content for site revision or reconstruction. For example, if the new page version is related to the “credit card”, then the web site keywords “credit, points and promotions” must be designed for the rewritten page text content.

9.5.2 Online recommendations

The idea is to use the discovered clusters, page specific data and rules to create a correct navigation recommendation. This process needs online session identification, therefore each session must implement a cookie mechanism.

The online navigation recommendations are created in the following way. User browsing behaviour is first classified by one of the discovered clusters. This is done by comparing the current navigation with the centroid by using a similarity measure introduced in (7.8). Let $\alpha = [(p_1, t_1), \dots, (p_m, t_m)]$ be the current user session and let $C_\alpha = [(p_1^\alpha, t_1^\alpha), \dots, (p_H^\alpha, t_H^\alpha)]$ be the centroid such as $\max\{sm(\alpha, C_i)\}$, with C_i the set of centroids discovered. The recommendations are created as a set of links to pages whose text content is related to p_{m+1}^α . These pages are selected with the business expert's collaboration.

Let $R_{m+1}(\alpha)$ be the online navigation recommendation for the $(m+1)^{th}$ page to be visited by user α , where $\delta < m < H$ and δ the minimum number of pages visited to prepare the suggestion. Then, we can write $R_{m+1}(\alpha) = \{l_{m+1,0}^\alpha, \dots, l_{m+1,j}^\alpha, \dots, l_{m+1,k}^\alpha\}$, with $l_{m+1,j}^\alpha$ the j^{th} link page suggested for the $(m+1)^{th}$ page to be visited by user α , and k the maximum number of pages for each suggestion. In this notation, $l_{i+1,0}^\alpha$ represents the “no suggestion” state.

As figure 9.1 shows, the CRS module is in charge of preparing the final set of links to be recommended to the user. For instance, if a user session is matched with cluster “1”, the most likely pages to be recommended are Products and Services, Promotions and Credit Cards.

9.5.3 Testing the recommendation effectiveness

The application of any recommendation will need the web site owner's agreement as some users may dislike the changes, which could become a potential risk for the

business. Unless carefully handled there is the danger that “*the cure might be worse than the disease*” and users migrate to other web sites.

In the case of a virtual bank and others where the web site is the core business, customer loss due to web site modifications can only be tolerated within a narrow range and only if it can be shown to retain existing customers and attract new customers in a short period of time. So the loss potential must be estimated by some a priori test, discussed in the following sections.

9.5.3.1 Testing offline structure recommendation

The main idea is to simulate user reactions to a new web site structure. Thus a secondary web site is created, whose page content is described in Table 9.10, based on the proposed changes and an usability test applied to measure reactions. Any web site structure, old and new, must help the user find what he or she is looking for; and this is the criterion for the usability test.

Table 9.10: New bank web site pages and their content

Pages	Contain
1	Home page
2, . . . , 70	Products and services
71, . . . , 105	Agreements with other institutions
106, . . . , 118	Remote services
119, . . . , 146	Credits cards
147, . . . , 155	Promotions
156, . . . , 180	Investments
181, . . . , 195	Different kinds of credits

The users can be grouped in two classes: experienced and inexperienced or “amateurs”. The latter is unfamiliar with a particular web site and possibly web technology. Their behavior is characterized by erratic browsing and sometimes they do not find what they are looking for. The former is users with site or other site experience and

with web technology. Their behavior is characterized by spending little time in pages with low interest and concentrating on the pages they are looking for and where they spend a significant amount of time.

As amateurs gain experience, they slowly become experienced users. Only experienced users are aware of a particular web site's features, so that recommendations for change should be based on their experience.

A usability test based on five web site visitors is regarded as enough for a site test. [173]. In this case, there are two amateur and three experienced users.

The experiment proceeded under the following conditions:

- Each user was asked to search in both general information and promotion for three different products (credit card, account, credit, etc), one of which was nonexistent. Next, the users had to write a simple description in three lines with the required information or 'not found', in the event they did not find what they were looking for.
- Each product information search is counted as a new session. It is required that the participant finishes the current session (using the finish session button).
- Participants used the same Internet connection, in this case, a Local Area Network.
- Because the information requested can be obtained in 4 or 5 clicks, it is considered as a "not found" or "lost in the hyperspace" status when the user visits 6 or more pages, even if the information was ultimately found.

Tables 9.11, 9.12 and 9.13 show the results of the experiments. The nomenclature used is "A" for amateur and "E" for experienced user respectively. The user's answers are given in the "Find information?" column. Those in parenthesis represent the real situation rather than the user's opinion.

The information obtained from the questions is found mainly in the following page ranges: 71 to 118 and 147 to 155. From Table 9.11, we see that user 1 spent considerable time on the pages without relevance to the question and did not find what he was looking for, although he thought he had. The other amateur user found the information, but had to visit five pages and spent time on pages with irrelevant information for the search purpose.

The experienced users found the information they were looking for.

Table 9.11: Navigation behavior searching the real web page A

#	user	Pages visited	Spent Time	Find information?
1	A	(1,4,15,18,72,79,...)	(3,50,8,30,4,5,...)	Yes(No)
2	A	(1,12,25,98,150)	(3,15,35,42,45)	Yes
3	E	(1,73,83,152)	(2,55,71,10)	Yes
4	E	(1,101,77,152)	(3,28,50,62)	Yes
5	E	(1,95,81,153)	(3,31,53,64)	Yes

The users in general, see Table 9.12, gained experience and all of them were able to find the requested information. Note that users tend to spend a significant time on pages whose content is related to the search purpose.

Table 9.12: Navigation behavior searching the real web page B

#	User	Visited Pages	Spent Time	Find information?
1	A	(1,75,82,148,154)	(2,20,40,75,42)	Yes
2	A	(1,116,94,118,154)	(2,40,28,65,42)	Yes
3	E	(1,75,108,147)	(2,50,41,67)	Yes
4	E	(1,82,87,151)	(3,35,43,50)	Yes
5	E	(1,84,149,150)	(3,49,45,62)	Yes

Finally, Table 9.13 shows the results for “the non existing product search” situation. The experienced user very quickly found that the relative information did not appear in the site. However, the amateur user tried to find it, looking in the site and

being confused by the answers.

Table 9.13: Navigation behavior searching a false web page

#	User	Visited pages	Spent Time	Find information?
1	A	(1,72,75,84,151,152,87,...)	(2,10,11,14,20,25,....)	Yes (No)
2	A	(1,110,78,150,146,155)	(2,6,10,8,9,2)	No
3	E	(1,104,105,76)	(2,6,5,4)	No
4	E	(1,95,153)	(3,6,8)	No
5	E	(1,102,147)	(2,4,4)	No

It can be concluded from this experiment that the users are able to find the information that they are searching for with a reduced number of visits and which is one of the objectives of the bank web site.

The second experiment consisted of a simple questionnaire to the same users about their impressions of the web site structure. The purpose was to understand how the new site structure helped users search for information. Table 9.14 shows the questionnaire results. Most users thought that the new web site hyperlink structure facilitated information search.

Table 9.14: Usability test for the web site hyperlink structure

#	Question	Acceptability opinion				
		Totally opposite	Moderately opposite	Some agree some opposite	Moderately agree	Totally agree
1	Does the site structure allows to find what is looking for?		1	1	3	
2	Does the site provide a consistent navigation?			1	4	
3	Does the site hides information?	2	2	1		
4	Is the site structure easy to understand?			1	3	1

The usability test provides some indication about the effectiveness of the proposed changes to the web site structure. However, the real test will be when the new web site version is released. These changes will be introduced gradually to avoid a “lost in hyper-space” feeling which could occur if all were made at the same time.

9.5.3.2 Testing offline content recommendation

As noted above web site keywords must be seen in context. To test the effectiveness of web site keywords, understood as the capacity to attract user attention during a web page session, a textual fragment such as a paragraph, should be created. These texts are a data source for web site keyword identification and although it is possible to use fictional examples, it was decided to use texts belonging to the web site itself as alternatives could unwittingly exaggerate attention to the test’s detriment. Therefore web texts were used so that a similar stylistic and information environment were maintained.

Five paragraphs were selected from the bank web site. Two contained the greatest number of web site keywords while the others were extracted randomly. All examples were shown to the same amateurs and expert users.

Table 9.15 shows the results of the web site keyword effectiveness test. The users showed a good receptivity toward paragraphs that contained the keywords, considering them interesting and with relevant information. So for the user, the particular words contain important information - the words, particularly web site keywords attract user attention.

Web site keywords can guide the web site designer about the specific text content. Of course, the utilization of the keywords does not guarantee the success of the paragraph, for it must be combined with elements such as semantic content, style and the paragraph meaning, all important for transmitting the message to the users.

Table 9.15: Testing the web site keyword effectiveness

#	Including the web site keyword?	Acceptability opinion				
		Irrelevant	Moderately irrelevant	Some information	Moderately relevant	Relevant
1	Yes				3	2
2	Yes			1	2	2
3	No	2	2	1		
4	No		3	2		
5	No		4	1		

9.5.3.3 Testing online navigation recommendation

The same web data as that found in the pattern discovery stage can be used to test the effectiveness of new initiatives. Patterns can be extracted from the complete web data and tests made as their effectiveness [236].

Let $ws = \{p_1, \dots, p_n\}$ be the web site and the pages that compose it. Using the distance introduced in (7.3) and with the collaboration of a web site content expert, we can define an equivalence class for the pages, where the pages belonging to the same class contain similar information. The classes partition the web site in disjoint subsets of pages.

Let Cl_x be the x^{th} equivalent class for the web site. It is such as $\forall p_z \in Cl_x, p_z \notin Cl_y, x \neq y \quad \bigcup_{x=1}^w Cl_x = ws$ where w is the number of equivalence classes. Let $\alpha = [(p_1, t_1), \dots, (p_H, t_H)]$ be a user behavior vector from the test set. Based on the first m pages actually visited, the proposed system recommends for the following $(m+1)$ page several possibilities, i.e., possible pages to be visited.

We test the effectiveness of the recommendations made for the $(m+1)^{th}$ page to be visited by user α following this procedure. Let Cl_q be the equivalence class for p_{m+1} . If $\exists l_{m+1,j}^\alpha \in R_{m+1}^\alpha / l_{m+1,j}^\alpha \in Cl_q, j > 0$ then we assume the recommendations were successful.

Although the set of link pages to be included could be larger, it could well lead

to confusion about which page to follow. We set in k the maximum number of pages per recommendation. Using the page distance introduced in (7.3), we can extract the closest k pages to p_{m+1} of the recommended changes.

$$E_{m+1}^k(\alpha) = \{l_{m+1,j}^\alpha \in \text{sort}_k(sp(p_{m+1}, l_{m+1,j}^\alpha))\}, \quad (9.1)$$

with sp the page distance introduced in (7.3). The “ sort_k ” function sorts the result of sp in descendent order and extracts the “ k ” link pages with the biggest distance to p_{m+1} . A particular case is when $E_{m+1}(\alpha) = \{l_{m+1,0}^\alpha\}$, i.e., no recommendation is proposed.

The above methodology was applied to the the bank web site data. They are activated when the user clicks the third page in their sessions. From the 30% of the user behavior vectors that belong to the test set, only those with six real components are selected, i.e. it was not necessary for complete user vectors with zeros to get the six components. Given this selection, we obtained 11,532 vectors to test the effectiveness of the online navigation suggestions.

Figure 9.12 shows the clusters identified using the 70% of the data. The centroid are presented in more detail in Table 9.16.

It should be noted that the clusters identified using the complete set of data and the 70% have similar pages contained in each centroid.

Table 9.16: User behavior clusters using 70% of the data

Cluster	Visited Pages	Time spent in seconds
1	(2,11,25,33,136,205)	(10,50,120,130,150,58)
2	(120,117,128,126,40,62)	(30,41,52,68,101,18)
3	(81,86,148,190,147,83)	(8,55,40,11,72,101)
4	(161,172,180,99,108,1)	(8,71,115,97,35,9)

After creating the navigation map as shown in Table 9.17, it is possible to un-

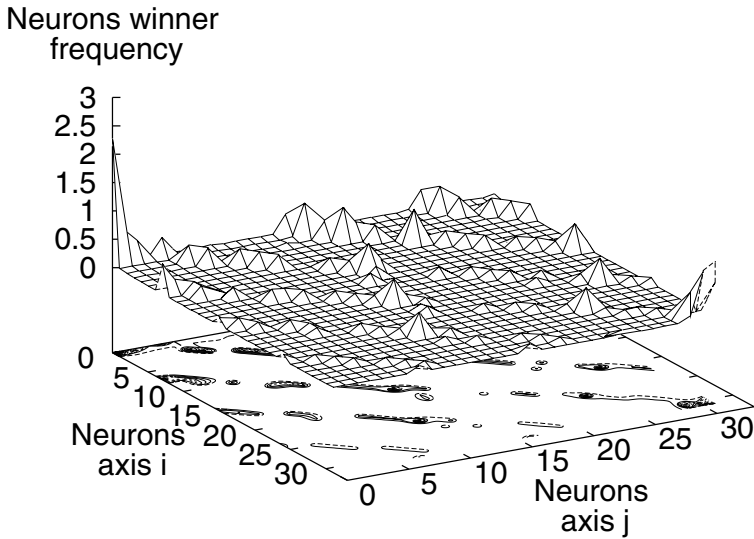


Figure 9.12: Clusters of user behavior vectors using 70% of the data

dertake the effectiveness experiment.

Figure 9.13 shows a histogram representing the percentage of the accepted suggestions using our validation method. As can be seen, acceptance increases if more pages are suggested for each page visit.

With this methodology, if only one page was suggested, it would have been accepted in slightly more than 50% of the cases. This was considered to be a very successful suggestion by the business expert, as the web site is complex with many pages, many links between pages, and a high rate of users that leave the site after a few clicks.

Furthermore, the probability of acceptance would have been even higher if the respective page had been suggested online during the session. Since we are comparing past visits stored in log files, we could only analyze the behavior of users unaware of the suggestions proposed.

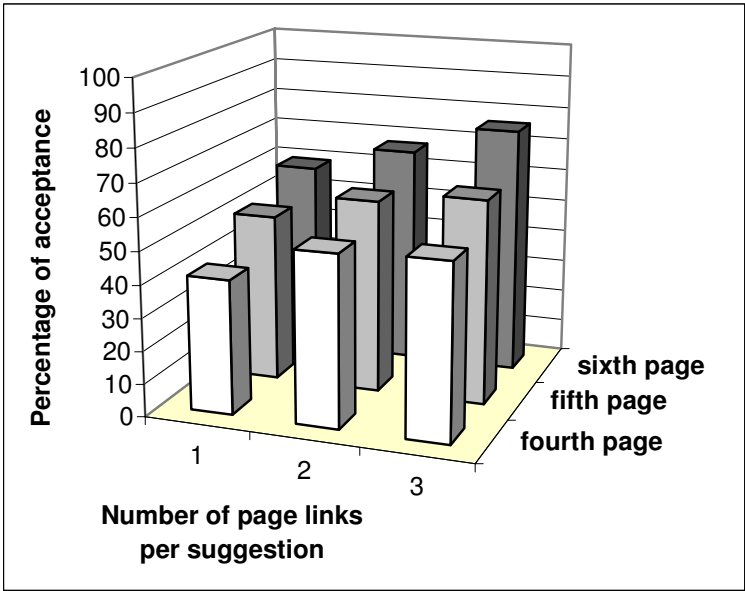


Figure 9.13: Percentage of acceptance of online navigation recommendations

9.5.4 Storing the extracted knowledge

The knowledge extracted from the bank web data is structured as patterns and rules for creating offline and online patterns in the Knowledge Base (see section 8.4.3). They are not expressed in computer language but as textual hints about how to use the patterns extracted in a way that is comprehensible to human users, like the web master and web site owners. Things are different in the case of online recommendations, where the user is a computerized recommender system and the rule must be written in a formal computer language. In both cases, the recommendations are enriched by using the information contained in the WIR.

9.5.4.1 Pattern Repository

The general structure of the Pattern Repository (PR) was presented in figure 8.4,. The measures in the fact table correspond to the pages recommended including the statistics about its use and the web site keywords for the period analyzed. These patterns are consulted using the information contained in the dimensional tables. Some examples are given below:

Time. (2003,Oct,4,26,18), i.e. “18:00 hours, October 26th, fourth week, year 2003”.

Text_preference. The **theme** column contains a short description about the main topic where the web site keywords should be used, for instance “*credit card and new customer promotion*”. In the **theme** column, it appears the range of date used for extracting the web site keywords, and in the **recommend** column some hints about possible paragraphs to be created, by using these keywords.

Browsing_Behavior. The cluster centroids discovered by the web mining process and shown in Table 9.5, as well as the formula in equation (7.8). Also, the offline recommendations to modify the web site hyperlink structure are contained in the **recommend** column.

WMT. It describes the web mining tool used, for example “*Self-organizing Feature Map with toroidal architecture*”, 32x32 neurons.

The table **Time** shows the date when the web mining tool was applied over the WIR. The dimensional table **Browsing_Behavior** stores the extracted patterns and suggestions about how to use them. The **WMT** contains information about the specific web mining tool applied, and in **Text_preferences** some hints about how to use the web site keywords in the fact table are listed.

A human user can query the PR, obtains suggestions that represent offline recommendations. The same query scheme could be used by an automatic system to prepare online navigation recommendations.

9.5.4.2 Rules for navigation recommendations

In order to prepare the online recommendation for the $(m + 1)^{th}$ page to visit, we compare the current session with the patterns in the Patterns Repository. A comparison needs a minimum of three pages visited ($\delta = 3$) to determine the cluster centroid most similar to the current visit and prepares a recommendation for the fourth page to be visited. This process can be repeated only after the user has visited more than three pages. For example, in the recommendations for the fifth page, we use the prior four pages visited from the current session.

The final online recommendation uses rules developed together with the domain expert. Sixteen rules were created for the four clusters found. We suggest at most three pages for the fourth, fifth, and sixth page to be visited, i.e., $k = 3$. A rule example is shown in Table 9.17. It belongs to the cluster “1”, for the fourth page recommendation. The *select* operation (lines 1 to 4) extracts the navigation recommendations and associate statistics from the pattern repository and compares them to the current user session, represented by the α parameter, with the navigation patterns, using the similarity measure set out in Eq. (7.8) (line 4). The result of the operation is stored in the “S” stack implemented by a simple linked list (line 1).

In this example, the patterns are clusters extracted using the SOFM. Let C_1 be the closet cluster (line 6) to α user (line 7), which is visiting the third page in this session and thus would like to recommend the fourth page to be visited.

The **Prepare_recommendation** function returns the “L” parameter which contains the the list of recommend links to pages. It is initialized in a “no recommendation” state (line 12). Next the “S” is reviewed for preparing the recommendation (lines 14 to 21). Before preparing the navigation recommendation, an edge condition is that pages to be recommend must belong to the current web site version (line 15), represented by the “ws” parameter (line 8). Sometimes this page belongs to an old site version. In those cases, the **Compare_page** extracts from the current site link to the page closest in content with the old page is recommended (line 16).

Table 9.17: An example of rule operation

1	select navigation, statistics into S
2	from pr_fact, time, browsing_behavior, wmt where “star join”
3	and “fix technique” and “fix time” and
4	sm(pattern,current_user) > ϵ ;
5	...
6	$C_1 \rightarrow [(1,40),(3,67),(8,175),(9,113),(147,184),(190,43)]$
7	$\alpha \rightarrow [(2,10),(11,50),(25,120)]$ % current user
8	ws $\rightarrow \{p_1, \dots, p_{217}\}$ % current web site pages
9	S.navigation $\rightarrow \{\{p_{33}, p_{38}, p_{41}, p_{118}, p_{157}, p_{201}\}$
10	S.statistic $\rightarrow \{1.2, 2.1, 1.8, 0.9, 0.8, 0.1\}$
11	Case C_1 and RecommPage=4 :
12	Prepare_recommendation($p_0, 0, L$); % default “no recommendation”
13	% L: link page recommendation, 0: statistic associated
14	while S not null loop
15	if (S.navigation not in ws) then
16	S.navigation = compare_page(ws,S.navigation);
17	elseif ((S.navigation <> $\alpha_{p_1 \dots 3}$) and (S.statistic > γ)) then
18	Prepare_recommendation(S.navigation,S.statistic,L);
19	Pop(S); % Next element in S
20	end if;
21	end loop;
22	send(Extracted_Three_Links(L)); % $L \rightarrow \{p_{38}, p_{41}, p_{33}\}$

Another edge condition is when the recommended page does not belong to the current α user session and when its associated statistics are greater than the γ parameter, which renders a minimum percentage of visits to the page interesting (line 17).

To process the following recommendation, the function **Pop(S)** extracts the next element in “S” (line 19). From the final set of pages in “L”, **Extracted_Three_Links** represents the expression (9.1) to extract a subset with a maximum of three links, based on the associated statistical data. The default is the no suggestion state. These links are sent (line 22) to the computerized recommender system which prepares the web page containing proposed navigation recommendations.

9.6 Summary

This chapter has presented a generic methodology for the application of web mining tools on web data to create an Adaptive Web Site (AWS). The methodology uses a Web Information Repository (WIR) and a Knowledge Base (KB) for storing both the information and the knowledge extracted from web data. The two structures have two specific users: human beings, such as webmasters, and artificial systems. The human users use the WIR and KB in order to get offline structural and content recommendations to modify the site. The artificial system is a Computerized Recommender System (CRS) and queries are executed in real time.

Information extraction tools and web mining algorithms are the principal applications used to track significant user behavior information and knowledge at a real and complex web site. In the example presented in this chapter, the case study site belonged to a bank and the web data refers to transactions over three months of continuous operations.

To create the WIR, a data mart based on the star query model was implemented to store the information originating from web data. The WIR is used by human beings and the information contained is used as an input to the web mining algorithms.

Cluster methods, from all available data mining techniques, were found to be the most fruitful for discovering significant user behavior patterns. Two clustering techniques were applied using the similarity measure proposed in Eq.(7.8). The first, the SOFM with a toroidal topology, maintains cluster continuity - a useful characteristic when processing data corresponding to a sequence of events like visitor behavior. The second technique is a simple version of K-means, which uses an alternative test for similarity measure.

One possible disadvantage of cluster identification is that it depends on subjective parameters. The cluster must be interpreted in terms of the appropriate business context.

When experienced customers visit a web site, there is a correlation between the maximum time spent per session in a page and its free text content. Hence the concept of “web site keyword” defined as a word or set of words that attract the visitor. These keywords then convey information about visitor web site preferences. The most important pages by session are identified by sorting visitor behavior vectors by time component. So the “Important Page Vector (IPV)” can be defined and a new similarity measure applied to find clusters among these vectors.

Clusters of IPVs were found by using SOFM and an extraction process using keywords. It uses the representation of the page in the vector space model which calculates the geometric media for the vectors in each cluster. The results are sorted and the most significant words are extracted.

Two structures, the Pattern and Rule Repositories, are used to construct the KB. The patterns discovered after applying the web mining tools are stored in the Pattern Repository. The Rule Repository contains knowledge about “how to use the patterns”, facilitating navigation recommendations.

As changes to a business web site could effect its core business, they must be made with caution. Thus even a-priori testing of the recommended changes should be measured for their effectiveness.

A usability test was applied to a representative sample of offline structural and content proposals. The first results showed that the recommended changes were not in the right direction. However, as noted, it is advisable to modify a site gradually, checking user reactions, but not all at once.

Our experiment tested the online navigation recommendations which used 70% of the web log registers to generate rules, and then provided online navigation suggestions based on the proposed system and respective domain knowledge. The rules were applied to the remaining 30% of the web log files, and the results indicated that if one page only was recommended, it would have been accepted by just over 50% of

the web users. A business expert thought this a successful experiment, since the web site is a complex one, with many pages, many links between pages, and a high visitor turnover rate, suggesting that many visitors leaving the site after a few clicks.

The crucial test however will come when the AWS is fully operational and interacting with real users. The a priori tests are a sensible way of implementing and gradually correcting site recommendations as well as improving visitor-site interactions.

In place of conclusions

*A conclusion is the place
where you got tired of thinking.*

Arthur Bloch

The future of web-based systems will be strongly influenced by their ability of adapting to the needs of their potential users. Web personalization represents a current approach that encompasses the algorithms, techniques and methods for personalizing the web user experience when visiting a particular web site.

Any model and method behind the web personalization needs a maximum understanding about the web user browsing behavior and preferences. The good news is that, by construction, the Web is the biggest opinion poll that can be applied to our web users. The interaction between the web users and the web site are stored in files (web logs, web pages, web site hyperlink structure, etc.), administered by the web server, in this way making it possible to analyse the web user behavior. The bad news is that it is not trivial to analyse these data sources, also called web data. But, the knowledge discovery from databases (KDD) approaches provide sound methodologies that can be used in the case of web data.

Web data considers a wide spectrum of formats, data types, data precision, etc. Before applying a pattern extraction process on web data, it is necessary to transform them into feature vectors, which represent the intrinsic variables of the phenomenon under study; in our case the web user behavior on the web site. This transformation

stage requires two important tasks to be previously done: data cleaning, for removing the data noise and errors, and data pre-processing, which selects the significant parameters to be considered in the feature vectors.

The feature vectors quality depends on the quality of the web data, and therefore any attempt to improving the data generation process is always very welcome. In this sense, the establishment of good practices for web data generation will be essential to any future analysis of the web users behavior on the web site. These good practices must consider adequate metadata for web objects in the web page, for instance how to provide more information about an image, sound, paragraph, etc. On the same lines, the web site structure must be documented, showing any changes during the site operation. Also the web logs registers must contain the maximum amount of data that the web server configuration can allow. A method for maintaining the web object versions, which allow one to know the version of the web object, is specified in a particular web log register, which is mandatory. All these practices aim to reduce the inherent noise in the web data, in order to generate data of high quality with the correct metadata associated, which will impact significantly on the whole pattern and knowledge discovery process and improve what we can learn about the web user.

The feature vector components depend on the phenomenon under study and what characteristics we want to review, requiring us to model the particular situation that is to be analysed. Hence, we need to model the web user behavior in a web site, by privileging the characteristics that we wish to analyse. Several web user behavior models can be created depending on who and what we want to analyse. These models will finally have a pragmatic representation in a feature vector, which will be used as input for a web-mining algorithm.

From the patterns extracted after applying a web mining process, we can find important facts related to the web user behavior in a particular web site and, with the support of the expert in the business under study, validate or reject these facts, which is a step back from representing the extracted knowledge in some way that is

to be used for improving the relationship between the web user and the web site.

Adaptive web sites (AWS) are emerging as the new portal generation, which are based exclusively on the web user behavior, changing the site structure and content in order to satisfy the needs of a particular web user. These changes are performed in two ways: offline and online. The former represent modifications in the site structure and content with the web server in shutdown mode. The latter represent navigation recommendations to the web user, in order to help him find what he is looking for in the web site. More complex adaptations, like changing the web pages content online, remain a big challenge, which is being tackled at the moment with promising results.

Although the intention behind any web-based system for assisting the web users in their search is always commendable, there is a subjacent problem related to the web user data privacy. Indeed, we should consider the fact that we may know so much about our digital customers, as there still remains the ethical question of accessing their personal information, which in many countries is prohibited. In this sense, the main idea is to learn about groups of users, and to refrain from identifying the personal data of each individual user one by one. We are analyzing enormous quantities of data in order to extract common patterns about the users navigation behavior and content preferences.

Due to the shifting nature of the web users needs and desires, the adaptation capacity of the web site to these new requirements will be the key to assuring the web site survival in the digital market. Because the AWS is based on the web user behavior, it is mandatory to develop methods for acquiring, maintaining and using the information and knowledge extracted from the web data. In this light, as argued across the chapters of this book, the Web Information Repository (WIR) based on the data warehousing architecture and the Knowledge Base (KB) represent a fundamental stage in the construction of an AWS. These structures must evolve in order to gradually store the changes in the Web. Today, we are facing the Web 2.0, which is incorporating semantic information about web objects that compound the web pages.

This new information will both improve and increase what we can learn about web users behavior in the Web, rendering the creation of real-world complex web-based systems possible, which for the moment is considered to be a problem of academic interest only.

Bibliography

- [1] K. Aas and L. Eikvil. Text categorisation: A survey. Technical report, Norwegian Computing Center, February 1999.
- [2] A. Abraham and V. Ramos. Web usage mining using artificial ant colony clustering and genetic programming. In *Procs. Int. Conf. CEC03 on Evolutionary Computation*, pages 1384–1391. IEEE Press, 2003.
- [3] G. Adomavicius and A. Tuzhilin. Personalization technologies: A process-oriented perspective. *Communications of ACM*, 48(10):83–90, October 2005.
- [4] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, June 2005.
- [5] P. Adriaans and D. Zantinge. *Data Mining*. Addison Wesley, Edinburgh, England, 1996.
- [6] R. Agrawal, A. Gupta, and S. Sarawagi. Association rules between sets of items in large databases. In *n Proc. of the ACM SIGMOD Conference on Management of Data*, pages 207–216, May 1993.
- [7] R. Agrawal, T. Imielinski, and A.S. Mining. Modeling multidimensional databases. In *Procs. 13th Int. Conf. Data Engineering ICDE*, pages 232–243, November 1997.
- [8] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Procs. of the 20th International Conference on Very Large Data Bases*, pages 487–499, June 1994.
- [9] E. Amitay and C. Paris. Automatically summarizing web sites: Is there any wayaround it? In *Procs. of the 9th Int. Conf. on Information and Knowledge Management*, pages 173–179, McLean, Virginia, USA, 2000.

- [10] D.M. Sow and D.P. Olshefski, M. Beigi, and G.S. Banavar. Prefetching based on web usage mining. In *Procs. Int. Conf. on IFIP/ACM/USENIX Middleware*, pages 262–281, 2003.
- [11] C. Apte, F. Damerau, and S. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, 2004.
- [12] L. Ardissono, L. Console, and I. Torre. An adaptive system for the personalized access to news. *AI Communications*, 14(3):129–147, 2001.
- [13] A. P. Asirvatham. Web page categorization based on document structure, 2003. Last accessed 9/13/2005.
- [14] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.
- [15] R. A. Baeza-Yates. *Web Mining: Applications and Techniques*, chapter Query Usage Mining in Search Engines, pages 307–321. Idea Group, 2004.
- [16] R. A. Baeza-Yates. Applications of web query mining. In *Proc. Int. Conf. ECIR*, pages 7–22, 2005.
- [17] R.A. Baeza-Yates. *Web mining and applications and techniques*, chapter Web usage mining in search engines. Idea Group, 2004.
- [18] G.B. Ball and D.J. Hall. A clustering technique for summarizing multivariate data. *Behavioral Science*, 12:153–155, 1967.
- [19] M. Batty. The computable city. *International Planning Studies*, 2(2):155–173, 1997.
- [20] N.J. Belkin. Helping people find what they don’t know. *Communications of the ACM*, 43(8):58–61, 2000.
- [21] B. Berendt, A. Hotho, and G. Stumme. Towards semantic web mining. In *Proc. in First Int. Semantic Web Conference*, pages 264–278, 2002.

- [22] B. Berendt, B. Mobasher, and M. Spiliopoulou. Web usage mining for e-business applications. Tutorial, ECMMML/PKDD Conference, August 2002.
- [23] B. Berendt and M. Spiliopoulou. Analysis of navigation behavior in web sites integrating multiple information systems. *The VLDB Journal*, 9:56–75, 2001.
- [24] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, and A. Secret. The world wide web. *Communications of ACM*, 37(8):76–82, 1994.
- [25] M.J.A. Berry and G. Linoff. *Data Mining Techniques*. Jon Wiley & Sons, New York, 1997.
- [26] M.J.A. Berry and G. Linoff. *Mastering Data Mining*. Jon Wiley & Sons, New York, 2000.
- [27] M.W. Berry, Z. Drmac, and E. R. Jessup. Matrices, vector spaces, and information retrieval. *SIAM Review*, 41(2):335–362, 1999.
- [28] D. Bertsimas and I. Popescu. Optimal inequalities in probability theory: A convex optimization approach. *SIAM J. on Optimization*, 15(3):780–804, 2005.
- [29] A. Bestavros. Using speculation to reduce server load and service time on the www. In *In Proc. 4th Int. Conf. ACM International Conference on Information and Knowledge Management*, pages 782–786, Baltimore, Maryland, USA, 1995.
- [30] S.S. Bhowmick, S.K. Madria, W.K. Ng, and E.P. Lim. Web warehousing: Design and issues. In *Procs. Int. of ER Workshops*, pages 93–104, 1998.
- [31] M. Boehnlein and A. Ulbrich vom Ende. Deriving initial data warehouse structures from the conceptual data models of the underlying operational information systems. In *Procs. Int. ACM Second International Workshop on Data Warehousing and OLAP*, pages 15–21, Kansas City, USA, November 2004.
- [32] F. Bonchi, F. Giannotti, C. Gozzi, G. Manco, M. Nanni, D. Pedreschi, C. Renso, and S. Ruggieri. Web log data warehousing and mining for intelligent web caching. *Data and Knowledge Engineering*, 32(2):165–189, 2001.

- [33] A. Bonifati, F. Cattaneo, S. Ceri, A. Fuggetta, and S. Paraboschi. Designing data marts for data warehouses. *ACM Transactions on Software Engineering Methodology*, 4:452–483, 2001.
- [34] D. Bonino, F. Corno, and G. Squillero. A real-time evolutionary algorithm for web prediction. In *Procs. IEEE/WIC Int. Conf. on Web Intelligence*, pages 139–145, Halifax, Canada, October 2003.
- [35] J. Borges and M. Levene. Data mining of user navigation patterns. In *of the Web Usage Analysis and User Proling Workshop*, pages 31–36, San Diego, USA, 1999.
- [36] C. Bouras and A. Konidaris. Web components: A concept for improving personalization and reducing user perceived latency on the world wide web. In *Proc. Int. Conf. on Internet Computing*, volume 2, pages 238–244, Las Vegas, Nevada, USA, June 2001.
- [37] K.B. Bøving and J. Simonsen. Http log analysis as an approach to study-ing the use of web-based information systems. *Scandinavian Journal of Information Systems*, 16:145–174, 2004.
- [38] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1984.
- [39] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.
- [40] P. Brusilovsky. Adaptive web-based system: Technologies and examples. Tutorial, IEEE Web Intelligence Int. Conference, October 2003.
- [41] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):1–47, 1998.
- [42] D. Buttler. A short survey of document structure similarity algorithms. In *Procs. Int. Conf. on Internet Computing*, pages 3–9, 2004.

- [43] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Seeing the whole in parts: text summarization for web browsing on handheld devices. In *Procs. Int. of the 10th Int Conf in World Wide Web*, pages 652–662, Hong Kong, 2001.
- [44] M. Cadoli and F. M. Donini. A survey on knowledge compilation. *AI Communications*, 10(3-4):137–150, 1997.
- [45] L. D. Catledge and J. E. Pitkow. Characterizing browsing behaviors on the world wide web. *Computers Networks and ISDN System*, 27:1065–1073, 1995.
- [46] J.M. Cavero, C. Costilla, E. Marcos, M. G. Piattini, and A. Sánchez. *Managing data mining technologies in organizations: techniques and applications*, chapter A multidimensional data warehouse development methodology, pages 188–201. Idea Group, 2004.
- [47] S. Chakrabarti. Data mining for hypertext: A tutorial survey. *SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining*, 1(2):1–11, 2000.
- [48] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Resource compilation by analyzing hyperlink structure and associated text. In *Procs. Int.Conf. World-Wide Web conference*, pages 65–74, Brisbane, Australia, 1998.
- [49] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Procs. Int.Conf. of the ACM SIGMOD*, pages 307–318, Seattle, WA, USA, 1998.
- [50] S. Chakrabarti, B.E. Dom, S.R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg. Mining the web’s link structure. *IEEE Computer*, 32(8), August 1999.
- [51] P.K. Chan. Constructing web user profiles: A non-invasive learning approach. In *WEBKDD '99: Revised Papers from the International Workshop on Web Usage Analysis and User Profiling*, pages 39–55, London, UK, 2000. Springer-Verlag.

- [52] G. Chang, M.J. Healey, J.A.M. McHugh, and J.T.L. Wang. *Mining the World Wide Web*. Kluwer Academic Publishers, 2003.
- [53] S. Chaudhuri and U. Dayal. An overview of data warehouse and olap technology. *SIGMOD Record*, 26(1):65–74, 1997.
- [54] M.S. Chen, J.S. Park, and P.S. Yu. Efficient data mining for path traversal patterns. *IEEE Trans. on Knowledge and Data Engineering*, 10(2):209–221, 1998.
- [55] T. Chenoweth, D. Schuff, and R. St. Louis. A method for developing dimensional data marts. *Communications of the ACM*, 46(12):93–98, 2003.
- [56] W.T. Chuang and J. Yang. Extracting sentence segment for text summarization? a machine learning approach. In *Procs. Int. Conf. ACM SIGIR*, pages 152–159, Athens, Greece, 2000.
- [57] E.F. Codd. Relational database: A practical foundation for productivity. *Communications of the ACM*, 25(2):109–117, 1982.
- [58] E.F. Codd. Providing olap (on-line analytical processing) to user-analysts: an it mandate. Technical report, 1993.
- [59] F. Coenen, G. Swinnen, K. Vanhoof, and G. Wets. A framework for self adaptive websites: tactical versus strategic changes. In *Procs. in 4th PAKDD Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 1–6, April 2000.
- [60] M.D. Cohen, C.B. Kelly, and A.L. Medaglia. Decision support with web-enabled software. *Information Systems-Decision Support Systems*, 31(2):109–129, 2001.
- [61] S. Colliat. Olap, relational and multidimensional database systems. *SIGMOD Record*, 25(3):64–69, 1996.

- [62] R. Cooley, B. Mobasher, and J. Srivastava. Grouping web page references into transactions for mining world wide web browsing patterns. In *Proc. in IEEE Knowledge and Data Engineering Workshop*, pages 2–9, November 1997.
- [63] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems*, 1:5–32, 1999.
- [64] R. W. Cooley. *Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data, Dissertation for degree of Doctor of Philosophy*. University of Minnesota, Faculty of the Graduate School, Minnesota, USA, 2000.
- [65] N. Correia and M. Boavida. Towards an integrated personalization framework: A taxonomy and work proposals. In *Workshop on Personalization in Electronic Publishing*, November 2001.
- [66] L. F. Cranor. 'i didn't buy it for myself': Privacy and ecommerce personalization. In *In Procs. of the Second ACM Workshop on Privacy in the Electronic Society*, pages 111–117, New York, USA, 2003.
- [67] A. Datta, K. Dutta, D. VanderMeer, K. Ramamritham, and S. B. Navathe. An architecture to support scalable online personalization on the web. *VLDB Journal: Very Large Data Bases*, 10(1):104–117, 2001.
- [68] R. Davis, H. Shrobe, and P. Szolovits. What is a knowledge representation. *AI Magazine*, 14(1):17–33, 1993.
- [69] B. D. Davison. Predicting web actions from html content. In *Procs. 13th ACM Int. Conf. on Hypertext and Hypermedia*, pages 159–168, 2002.
- [70] B.D. Davison. A web caching primer. *IEEE Internet Computing*, 5(4):38–45, Julyr 2001.
- [71] B.D. Davison, D.G. Deschenes, and D.B. Lewanda. Finding relevant website queries. In *In Poster Proceedings of the Twelfth Int. World Wide Web Conf.*, 2003.

- [72] J.K. Debenham. Knowledge base maintenance through knowledge representation. In *Procs. 12th Int. Conf. on Database and Expert Systems Applications*, pages 599–608, München, Germany, September 2001.
- [73] F. Dellmann, H. Wulff, and S. Schmitz. Statistical analysis of web log files of a german automobile producer. Technical report, Fachhochschule Münster, University of Applied Sciences, February 2004.
- [74] P. Denning. Electronic junk. *Communications of the ACM*, 25(3):163–165, March 1982.
- [75] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions On Information Systems*, 22(1):143–177, 2004.
- [76] V. Devedzic. Knowledge discovery and data mining in databases. Technical report, School of Business Administration, University of Belgrade, Yugoslavia, 2002.
- [77] D. Dhyani, W.K. Ng, and S.S. Bhowmick. A survey of web metrics. *ACM Computing Surveys*, 34(4):469–503, 2002.
- [78] M. Dikaiakos, A. Stassopoulos, and L. Papageorgiou. Characterizing crawler behavior from web server access logs. *Lecture Notes in Computer Science*, 2738(1):369–/378, 2003.
- [79] D. Duchamp. Prefetching hyperlinks. In *Procs. of the 2nd Intl. USENIX Symposium on Internet Technologies and Systems (USITS'99)*, Boulder, CO, October 1999.
- [80] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, NY, USA, 2000.
- [81] B. Dunkel, N. Soparkar, J. Szaro, and R. Uthurusamy. Systems for KDD: From concepts to practice. *Future Generation Computer Systems*, 13(2–3):231–242, 1997.

- [82] Oracle Education. Oracle 9i administration guide. Manual, Oracle Corporation, <http://otn.oracle.com>, 2002.
- [83] M. Eirinaki and M. Vazirgannis. Web mining for web personalization. *ACM Transactions on Internet Technology*, 3(1):1–27, 2003.
- [84] A. Famili, W.M. Shen, R. Weber, and E. Simoudis. Knowledge discovery in databases: An overview. *Data Preprocessing and Intelligent Data Analysis*, 1(1):3–23, 1997.
- [85] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. *Ai Magazine*, 17:37–54, 1996.
- [86] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Knowledge discovery and data mining: Towards a unifying framework. In *Procs. of Second Int. Conf. on Knowledge Discovery and Data Mining*, pages 82–88, 1996.
- [87] R. Feldman and I. Dagan. Knowledge discovery in textual databases (kdt). In *Procs Int. Conf. First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, pages 112–117, Montreal, Canada, 1995.
- [88] D.H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.
- [89] D.H. Fisher. Iterative optimization and simplification of hierarchical clusterings. *Journal of Artificial Intelligence Research*, pages 147–179, 1996.
- [90] G. Flake, S. Lawrence, C. Giles, and F Coetzee. Self-organization of the web and identification of communities. *Computer*, 35(3):66–71, 2002.
- [91] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–160, New York, NY, USA, 2000. ACM Press.

- [92] L.R. Ford, D.R. Fulkerson, and R.L. Rivest. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.
- [93] I. Foster and C. Kesselman. *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
- [94] W.J. Frawley, G. Piatetsky-Shapiro, and C.J. Matheus. Knowledge discovery in databases: An overview. *AI Magazine*, pages 57–70, 1992.
- [95] J. Gallaughier and S. Ramanathan. Choosing a client/server architecture. a comparison of two-tier and three-tier systems. *Information Systems Management Magazine*, 13(2):7–13, 1996.
- [96] D. Gefen. Customer loyalty in e-commerce. *Journal of the Association for Information Systems*, 3:27–51, 2002.
- [97] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proc. 9th ACM Conference on Hypertext and Hypermedia*, pages 225–234, 1998.
- [98] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, December 1992.
- [99] M. Golfarelli and S. Rizzi. Designing the data warehouse: key steps and crucial issues. *Journal of Computer Science and Information Management*, 2(1):1–14, 1999.
- [100] M. Gyssens and L.V.S. Lakshmanan. A foundation for multi-dimensional databases. In *The VLDB Journal*, pages 106–115, Bombay, India, 1997.
- [101] S.H. Ha. Helping online customers decide through web personalization. *IEEE Intelligent Systems*, 17(6):34–43, 2002.
- [102] P. R. Hagen, H. Manning, and R. Souza. Smart personalization. The forrester report, Forrester Research, Inc., Cambridge, MA, USA, July 1999.

- [103] U. Hahn and I. Mani. The challenges of automatic summarization. *IEEE Computer*, 33(11):29–36, 2000.
- [104] E.H. Han, G. Karypis, V. Kumar, and B. Mobasher. Clustering based on association rule hypergraphs. In *In Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 9–13, Tucson, Arizona, USA, 1997.
- [105] J.A. Hartigan and M.A. Wong. A k-means clustering algorithm. *Journal of the Applied Statistics*, 28:100–108, 1979.
- [106] B. Hay, G. Wets, and K. Vanhoof. Mining navigation patterns using a sequence alignment method. *Knowledge and Information Systems*, 6(2):150–163, 2004.
- [107] A. Hinneburg and D.A. Keim. Advances in clustering and applications. Tutorial, ICDM Int. Conf., Melbourne, Florida, USA, November 2003.
- [108] T. Hofmann. The cluster-abstraction model: Unsupervised learning of topic hierarchies from text data. In *of the 16th International Joint Conference on Artificial Intelligence IJCAI-99*, pages 682–687, 1999.
- [109] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen. Websom - self-organizing maps of document collections. In *In Proc. of Workshop on Self-Organizing Maps WSOM'97*, pages 310–315, 1997.
- [110] X. Hu and N. Cercone. A data warehouse/online analytic processing framework for web usage mining and business intelligence reporting. *International Journal of Intelligent Systems*, 19(7):585–606, 2004.
- [111] Z. Huang, J. Ng, D.W. Cheung, M.K. Ng, and W. Ching. A cube model for web acces sessions and cluster analysis. In *Proc. of WEBKDD*, pages 47–57, 2001.
- [112] C. A. Hurtado, A. O. Mendelzon, and A. A. Vaisman. Updating olap dimensions. In *International Workshop on Data Warehousing and (OLAP)*, pages 60–66, 1999.

- [113] W. H. Inmon. *Building the data warehouse (2nd ed.)*. John Wiley and Sons, New York, 1996.
- [114] B. Ives and G.P. Learmonth. The information system as a competitive weapon. *Communications of the ACM archive*, 27(12):1193–1201, 1984.
- [115] M. Iwayama and T. Tokunaga. Cluster-based text categorization: a comparison of category search strategies. In *In Procs. of the 18th International ACM SIGIRConference on Research and Development in Information Retrieval*, pages 273–280, Seattle, US, 1995.
- [116] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computer Survey.*, 31(3):264–323, 1999.
- [117] J. Jang, C. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, 1997.
- [118] J. Ji, Z. Sha, C. Liu, and N. Zhong. Online recommendation based on customer shopping model in e-commerce. In *Procs. IEEE/WIC Int. Conf. on Web Intelligence*, pages 68–74, Halifax, Canada, 2003.
- [119] Z. Jiang, A. Joshi, R. Krishnapuram, and L. Yi. Retriever: Improving web search engine results using clustering. Technical report, CSEE Department, UMBC, 2000.
- [120] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 137–142, London, UK, 1998. Springer-Verlag.
- [121] T. Joachims, D. Freitag, and T. Mitchell. Webwatcher: A tour guide for the world wide web. In *Procs. of the Fifteenth Int. Conf. Joint Conf. on Artificial Intelligence*, pages 770–775, 1997.
- [122] A. Johansen and D. Sornette. The nasdaq crash of april 2000: Yet another example of log-periodicity in a speculative bubble ending in a crash. *European*

Physical Journal B - Condensed Matter and Complex Systems, 17(2):319–328, September 2000.

- [123] A. Joshi and R. Krishnapuram. On mining web access logs. In *Proc. of the 2000 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 63–69, 2000.
- [124] A.R. Pereira Jr and N. Ziviani. Retrieving similar documents from the web. *Journal of Web Engineering*, 2(4):247–261, 2004.
- [125] G. Karypis, E.-H. E.H. Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *IEEE Computer*, 32(8):68–75, 1999.
- [126] M. Kilfoil, A. Ghorbani, W. Xing, Z. Lei, J. Lu, J. Zhang, and X. Xu. Toward an adaptive web: The state of the art and science. In *Procs. Annual Conference on Communication Networks & Services Research*, pages 119–130, Moncton, Canada, May 2003.
- [127] W. Kim. Personalization: Definition, status, and challenges ahead. *Journal of Object Technology*, 1(1):29–40, 2002.
- [128] R. Kimball and R. Merx. *The Data Webhouse Toolkit*. Wiley Computer Publisher, New York, 2000.
- [129] R. Kimball, L. Reeves, W. Thornthwaite, M. Ross, and W. Thornwaite. *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses*. John Wiley & Sons, Inc., New York, NY, USA, 1998.
- [130] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of ACM*, 46(5):604–632, 1999.
- [131] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins. The web as a graph: Measurements, models and methods. *Lecture Notes Computer Science*, 1627(1):1–17, 1999.

- [132] A. Kobsa. Tailoring privacy to users' needs. In *In Procs. of the 8th International Conference in User Modeling*, pages 303–313, 2001.
- [133] A. Kobsa, J. Koenemann, and W. Pohl. Personalized hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review*, 16(2):111–155, 2004.
- [134] Y. Kodratoff. Technical and scientific issues of kdd (or: Is kdd a science?). In *Proc. Int Conf. Algorithmic Learning Theory*, volume 997, pages 261–265, Fukuoka, Japan, October 1995.
- [135] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, 1987.
- [136] R. Kosala and H. Blockeel. Web mining research: A survey. *SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining*, 2(1):1–15, 2000.
- [137] P. Kotler and G. Armstrong. *Marketing: an introduction*. Prentice Hall, Upper Saddle River, New Jersey, 2000.
- [138] M. Koutri, N. Avouris, and S. Daskalaki. *Adaptable and Adaptive Hypermedia Systems*, chapter A survey on web usage mining techniques for web-based adaptive hypermedia systems. Idea Publishing Inc., Hershey, PA, USA, 2004.
- [139] M. Koutri, S. Daskalaki, and N. Avouris. Adaptive interaction with web sites: an overview of methods and techniques. In *Proc. of the 4th Int. Workshop on Computer Science and Information technologies, CSIT 2002*, Patras, Greece, 2002.
- [140] A. Kraiss and G. Weikum. Integrated document caching and prefetching in storage hierarchies based on markov-chain predictions. *VLDB Journal*, 7:141–162, 1998.
- [141] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web and social networks. *Computer*, 35(11):32–36, 2002.

- [142] S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Recommendation systems: A probabilistic analysis. In *Procs. IEEE Symposium on Foundations of Computer Science*, pages 664–673, 1998.
- [143] O. Kwon and J. Lee. Text categorization based on k-nearest neighbor approach for web site classification. *Information Processing & Management*, 39(1):25–44, 2003.
- [144] K. Lagus, S. Kaski, and T. Kohonen. Mining massive document collections by the websom method. *Information Sciences*, 163(1-3):135–156, 2004.
- [145] D. Lawrie, B. W. Croft, and A. Rosenberg. Finding topic words for hierarchical summarization. In *Proc. 24th Int SIGIR Conf. on Research and Development in Information Retrieval*, pages 349–357, New Orleans, Louisiana, USA, 2001. ACM Press.
- [146] J. Lee and W. Shiu. An adaptive website system to improve efficiency with web mining techniques. *Advanced Engineering Informatics*, 18(3):129–142, 2004.
- [147] M. Levene and G. Loizou. Why is the snowflake schema a good data warehouse design? *Information Systems*, 28:225–240, 2003.
- [148] V.I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys. Dokl.*, pages 705–710, 1966.
- [149] D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *In Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 37–50, Kobenhavn, DK, 1992.
- [150] Q. Li and R. Khosla. An adaptive e-commerce personalization framework with application in e-banking. *Lecture Notes in Computer Science*, 2822(1):16–26, 2003.

- [151] S. Li, J.C. Liechty, and A. Montgomery. Modeling category viewership of web users with multivariate count models. Working paper #2003-e25, GSIA, Business School, Carnegie Mellon University, July 2002.
- [152] E.D. Liddy, K. McVearry, W. Paik, E. Yu, and M. McKenna. Development, implementation and testing of a discoursemodel for newspaper texts. In *Procs. Int. Conf. on ARPA Workshop on Human Language Technology*, pages 159–164, Princeton, NJ, USA, 1993.
- [153] G. Linoff and M.J.A. Berry. *Mining the Web*. Jon Wiley & Sons, New York, 2001.
- [154] S. Loh, L. Wives, and J. P. M. de Oliveira. Concept-based knowledge discovery in texts extracted from the web. *SIGKDD Explorations*, 2(1):29–39, 2000.
- [155] Z. Lu, Y.Y. Yao, and N. Zhong. *Web Intelligence*. Springer-Verlag, Berlin, 2003.
- [156] I. Mani and M.T. Maybury. *Advances in automatic text summarization*. MIT Press, Cambridge, Mass., 1999.
- [157] T. Martyn. Reconsidering multi-dimensional schemas. *SIGMOD Record*, 33(1):83–88, 2004.
- [158] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *In AAAI-98 Workshop on Learning for Text Categorization*, pages 349–357, 1998.
- [159] M.L. Meuter, A.L. Ostrom, R. I. Roundtree, and M. J. Bitner. Self-service technologies: Understanding customer satisfaction with technology-based service encounters. *Journal of Marketing*, 64(3):50–64, July 2000.
- [160] P. Mika. Social networks and the semantic web. In *WI-2004 In Procs. Int. Conf. of the IEEE/WIC/ACM on Web Intelligence*, pages 285–291. IEEE Press, 2004.
- [161] S. Mitra, S. K. Pal, and P. Mitra. Data mining in soft computing framework: A survey. *IEEE Transactions on Neural Networks*, 13(1):3–14, 2002.

- [162] B. Mobasher, B. Berendt, and M. Spiliopoulou. Kdd for personalization. Tutorial, KDD Conference, September 2001.
- [163] B. Mobasher, R. Cooley, and J. Srivastava. Creating adaptive web sites through usage-based clustering of urls. In *Procs. Int Conf IEEE Knowledge and Data Engineering Exchange*, November 1999.
- [164] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.
- [165] B. Mobasher, T. Luo, Y. Sung, and J. Zhu. Integrating web usage and content mining for more effective personalization. In *Procs. of the Int. Conf. on E-Commerce and Web Technologies*, pages 165–176, September, Greenwich, UK, 2000.
- [166] D. L. Moody and M. A. R. Kortink. From enterprise models to dimensional models: A methodology for data warehouse and data mart design. In *Proceedings of the 2nd Intl. Workshop DMDW'2000*, pages 1–11, Stockholm, Sweden, June 2000.
- [167] B. Mortazavi-Asl. Discovering and mining user web-page traversal patterns. Master's thesis, Computing Science, Simon Fraser Univ., Canada, 2001.
- [168] K.R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, March 2001.
- [169] M.D. Mulvenna, S.S. Anand, and A.G. Buchner. Personalization on the net using web mining. *Communication of ACM*, 43(8):123–125, August 2000.
- [170] G.J. Myatt. *Making Sense of Data*. Willey-Interscience, 2007.
- [171] D.S.W. Ngu and X. Wu. Sitehelper: A localized agent that helps incremental exploration of the world wide web. *Computer Networks and ISDN Systems: The International Journal of Computer and Telecommunications Networking*, 29(8):1249–1255, 1997.

- [172] J. Nielsen. User interface directions for the web. *Communications of ACM*, 42(1):65–72, 1999.
- [173] J. Nielsen. Quantitative studies: How many users to test? Report, Usable Information Technology, 2006. Also available as http://www.useit.com/alertbox/quantitative_testing.html.
- [174] V.N. Padmanabhan and J.C. Mogul. Using predictive prefetching to improve world wide web latency. *ACM SIGCOMM Computer Communication Review*, 26(3):22–36, July 1996.
- [175] L. Page, S. Bring, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Tech. rep., Computer Systems Laboratory, Stanford University, Stanford, CA, USA, 1998.
- [176] S. K. Pal, V. Talwar, and P. Mitra. Web mining in soft computing framework: Relevance, state of the art and future directions. *IEEE Transactions on Neural Networks*, 13(5):1163–1177, September 2002.
- [177] D. Patterson, G. Gibson, and R. Katz. A case for redundant array of inexpensive disks (raid). In *Proc. Int Conf. ACM SIGMOD*, pages 109–116, June 1998.
- [178] M. J. Pazzani and D. Billsus. Adaptive web site agents. In Oren Etzioni, Jörg P. Müller, and Jeffrey M. Bradshaw, editors, *Procs. of the Third Int. Conf. on Autonomous Agents (Agents'99)*, pages 394–395, Seattle, WA, USA, 1999. ACM Press.
- [179] M. Perkowitz. *Adaptive Web Site: Cluster Mining and Conceptual Clustering for Index Page Synthesis*, Dissertation for degree of Doctor of Philosophy. University of Washington, 2001.
- [180] M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically synthesizing web pages. In *In Procs. of the 15th National Conference on Artificial Intelligence*, pages 727–732, Wisconsin, USA, July 1998.

- [181] M. Perkowitz and O. Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118(1-2):245–275, April 2000.
- [182] S. Perugini and M. A. Goncalves. Recommendation and personalization: a survey. Technical report cs.ir/0205059, Department of Computer Science, Virginia Tech, Available at <http://xxx.lanl.gov/abs/cs.IR/0205059>, 2002.
- [183] D. Pierrakos, G. Paliouras, C. Papatheodorou, and C. D. Spyropoulos. Web usage mining as a tool for personalization: A survey. *User Modeling and User-Adapted*, 13:311–372, 2003.
- [184] B. J. Pine. *Mass Customization*. Harvard Business School Press, Boston, USA, 1993.
- [185] M. F. Porter. An algorithm for suffix stripping. *Program; automated library and information systems*, 14(3):130–137, 1980.
- [186] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [187] E. Rasmussen. *Information retrieval: data structures and algorithms*, chapter Clustering algorithms, pages 419–442. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.
- [188] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, March 1997.
- [189] D. Riecken. Personalized views of personalization. *Communications of the ACM*, 43(8):27–28, 2000.
- [190] S.A. Ríos, J.D. Velásquez, E. Vera, and H. Yasuda. Establishing guidelines on how to improve the web site content based on the identification of representative pages. In *In Procs. IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology*, pages 284–288, September 2005.
- [191] S.A. Ríos, J.D. Velásquez, E. Vera, and H. Yasuda. Improving the web text content by extracting significant pages into a web site. In *In Procs. 5th IEEE Int.*

- Conf. on Intelligent Systems Design and Applications*, pages 32–36, September 2005.
- [192] S.A. Ríos, J.D. Velásquez, H. Yasuda, and T. Aok. Using a self organizing feature map for extracting representative web pages from a web site. *International Journal of Computational Intelligence Research*, 1(2):159–16, 2006.
- [193] S.A. Ríos, J.D. Velásquez, H. Yasuda, and T. Aoki. A hybrid system for concept-based web usage mining. *International Journal of Hybrid Intelligent Systems*, 3(4):219–235, 2006.
- [194] F. Roseblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington, DC, 1962.
- [195] T. A. Runkler and J. Bezdek. Web mining with relational clustering. *International Journal of Approximate Reasoning*, 32(2-3):217–236, Feb 2003.
- [196] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, Feb 1988.
- [197] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, uSA, 1983.
- [198] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM archive*, 18(11):613–620, November 1975.
- [199] C. Sapia, G. Höfling, C. Hausdorf, M. Müller, H. Stoyan, and U. Grimmer. On supporting the data warehouse design by data mining techniques. In *In GI-Workshop: Data Mining und Data Warehousing*, Magdeburg, Germany, September 1999.
- [200] J. B. Schafer, J. A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5:115–153, 2001.
- [201] H. Schutze, D. A. Hull, and J. O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *In Proceedings of SIGIR-95*,

18th ACM International Conference on Research and Development in Information Retrieval, pages 229–237, Seattle, US, 1995.

- [202] E. Schwarzkopf. An adaptive web site for the um 2001 conference. In *Procs. UM2001 Workshop on User Modeling, Machine Learning and Information Retrieval*, pages 77–86, November 2001.
- [203] F. Sebastiani. A tutorial on automated text categorisation. In *Procs. of ASAI-99, 1st Argentinean Symposium on Artificial Intelligence*, pages 7–35, Buenos Aires, Argentina, 1999.
- [204] A. Sen and A.P. Sinha. A comparison of data warehousing methodologies. *Communications of the ACM*, 48(3):79–84, 2005.
- [205] J. Shavlik and G. G. Towell. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1/2):119–165, 1994.
- [206] B.G. Silverman. Implications of buyer decision theory for design of e-commerce web sites. *Int. J. Human-Computer Studies*, 55(5):815–844, 2001.
- [207] M. Spiliopoulou. Data mining for the web. In *Principles of Data Mining and Knowledge Discovery*, pages 588–589, 1999.
- [208] M. Spiliopoulou, L.C. Faulstich, and K. Winkler. A data miner analyzing the navigational behaviour of web users. In *Procs. of workshop on Machine Learning in User Modeling of the ACAI'99*, pages 588–589, 1999.
- [209] M. Spiliopoulou, B. Mobasher, B. Berendt, and M. Nakagawa. A framework for the evaluation of session reconstruction heuristics in web-usage analysis. *INFORMS Journal on Computing*, 15:171–190, 2003.
- [210] J. Srivastava, R. Cooley, M. Deshpande, and P. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.

- [211] S. Staab, P. Domingos, P. Mika, J. Golbeck, L. Ding, T. Finin, A. Joshi, A. Nowak, and R.R. Vallacher. Social networks applied. *IEEE Intelligent Systems*, 20(1):80–93, 2005.
- [212] A. Strehl and J. Ghosh. Value-based customer grouping from large retail datasets. In *In Procs. of SPIE Conf. on Data Mining and Knowledge Discovery*, pages 33–42, Orlando, Florida, USA, 2000.
- [213] A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *Workshop on Artificial Intelligence for Web Search(AAAI 2000)*, pages 58–64, 2000.
- [214] L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter. Phoaks: A system for sharing recommendations. *Communications of the ACM*, 40(3):59–62, 1997.
- [215] O. Teste. Towards conceptual multidimensional design in decision support systems. In *Fifth East-European Conf. on Advances in Databases and Information Systems*, pages 25–28, Vilnius, Lithuania, 2001.
- [216] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 1999.
- [217] A. Thor, N. Golovin, and E. Rahm. Awesome - a data warehouse-based system for adaptive website recommendations. In *In Proc. of Int. Conf VLDB*, pages 384–395, 2004.
- [218] P. Tonella, F. Ricca, E. Pianta, and C. Girardi. Recovering traceability links in multilingual web sites. In *Procs. Int. Conf. Web Site Evolution*, pages 14–21. IEEE Press, 2001.
- [219] P. Tonella, F. Ricca, E. Pianta, and C. Girardi. Restructuring multilingual web sites. In *Procs. Int. Conf. Software Maintenance*, pages 290–299. IEEE Press, 2002.
- [220] P.D. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2:303–336, 2000.

- [221] V. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Nauka, Moscow, Russia, 1979.
- [222] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, USA, 1995.
- [223] P. Vassiliadis and T. K. Sellis. A survey of logical models for olap databases. *SIGMOD Record*, 28(4):64–69, 1999.
- [224] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos. On the logical modeling of etl processes. In *Proc. 14th Int. Conf. on Advanced Information Systems Engineering*, pages 782–786, London, UK, 2002.
- [225] Ch. Vassiliou, D. Stamoulis, and D. Martakos. The process of personalizing web content: techniques, workflow and evaluation. In *Procs Int. Conf. on Advances in Infrastructure for Electronic Business, Science and Education on the Internet*, 2002.
- [226] Ch. Vassiliou, D. Stamoulis, A. Spiliotopoulos, and D. Martakos. *Creating adaptive web sites using personalization techniques: a unified, integrated approach and the role of evaluation*, pages 261–285. Idea Group Publishing, Hershey, PA, USA, 2003.
- [227] J. D. Velásquez. *A Study on Intelligent Web Site: Towards a New Generation of Adaptive Web Portals*. PhD thesis, University of Tokyo, Japan, December 2004.
- [228] J. D. Velásquez, R. Weber, H. Yasuda, and T. Aoki. A methodology to find web site keywords. In *Procs. IEEE Int. Conf. on e-Technology, e-Commerce and e-Service*, pages 285–292, Taipei, Taiwan, March 2004.
- [229] J. D. Velásquez, H. Yasuda, and T. Aoki. Combining the web content and usage mining to understand the visitor behavior in a web site. In *Procs. 3th IEEE Int. Conf. on Data Mining*, pages 669–672, Melbourne, Florida, USA, November 2003.

- [230] J. D. Velásquez, H. Yasuda, and T. Aoki. Using self organizing map to analyze user browsing behavior. In *Procs. Int. Conf. Computer, Communication and Control Technologies*, volume 4, pages 64–68, Orlando, Florida, USA, August 2003.
- [231] J. D. Velásquez, H. Yasuda, T. Aoki, and R. Weber. Voice codification using self organizing maps as data mining tool. In *Procs. of 2th Int. Conf. on Hybrid Intelligent Systems*, pages 480–489, Santiago, Chile, December 2002.
- [232] J. D. Velásquez, H. Yasuda, T. Aoki, and R. Weber. *Data Mining*, chapter A generic Data Mart architecture to support Web mining, pages 389–399. Wit Press, Ashurst, Southampton, UK, 2003.
- [233] J. D. Velásquez, H. Yasuda, T. Aoki, and R. Weber. A new similarity measure to understand visitor behavior in a web site. *IEICE Transactions on Information and Systems, Special Issues in Information Processing Technology for web utilization*, E87-D(2):389–396, February 2004.
- [234] J. D. Velásquez, H. Yasuda, T. Aoki, R. Weber, and E. Vera. Using self organizing feature maps to acquire knowledge about user behavior in a web site. In *Procs. 7th Int. Conf. Knowledge-Based Intelligent Information & Engineering Systems*, volume 1, pages 951–958, Oxford, UK, September 2003.
- [235] J. D. Velásquez, H. Yasuda, T. Aoki, R. Weber, and E. Vera. Using self organizing feature maps to acquire knowledge about visitor behavior in a web site. *Lecture Notes in Artificial Intelligence*, 2773(1):951–958, September 2003.
- [236] J.D. Velásquez, , and V. Palade. Testing online navigation recommendations in a web site. *Lecture Notes in Artificial Intelligence*, 4253(1):487–496, 2006.
- [237] J.D. Velásquez, A. Bassi, H. Yasuda, and T. Aoki. Mining web data to create online navigation recommendations. In *Procs. 4th IEEE Int. Conf. on Data Mining*, pages 551–554, Brighton, UK, November 2004.

- [238] J.D. Velásquez, P.A. Estévez, H. Yasuda, T. Aoki, and E. Vera. Intelligent web site: Understanding the visitor behavior. *Lecture Notes in Computer Science*, 3213(1):140–147, 2004.
- [239] J.D. Velásquez and J. I. Fernández. Towards the identification of important-words from the web user point of view. In *Procs. on Int. Workshop on Intelligent Web Based Tools (IWB-T-07)*, CEUR-WS database, pages 17–26, Patras, Greece, October 2007.
- [240] J.D. Velásquez and V. Palade. Building a knowledge base for implementing a web-based computerized recommendation system. *International Journal of Artificial Intelligence Tools*, 16(5):793–828, 2007.
- [241] J.D. Velásquez and V. Palade. A knowledge base for the maintenance of knowledge extracted from web data. *Journal of Knowledge-Based Systems (Elsevier)*, 20(3):238–248, 2007.
- [242] J.D. Velásquez, S. Ríos, A. Bassi, H. Yasuda, and T. Aoki. Towards the identification of keywords in the web site text content: A methodological approach. *International Journal of Web Information Systems*, 1(1):11–15, March 2005.
- [243] J.D. Velásquez, R. Weber, H. Yasuda, and T. Aoki. Acquisition and maintenance of knowledge for web site online navigation suggestions. *IEICE Transactions on Information and Systems*, E88-D(5):993–1003, May 2005.
- [244] J.D. Velásquez, H. Yasuda, and T. Aoki. Web site structure and content recommendations. In *Procs. 3th IEEE/WIC Int. Conf. on Web Intelligence*, pages 636–639, Beijing, China, September 2004.
- [245] J.D. Velásquez, H. Yasuda, T. Aoki, and R. Weber. Using the kdd process to support the web site reconfiguration. In *Procs. IEEE/WIC Int. Conf. on Web Intelligence*, pages 511–515, Halifax, Canada, October 2003.

- [246] J. Wang and J. Lin. Are personalization systems really personal? effects of conformity in reducing information overload. In *Procs. of the 36th Hawaii Int. Conf. on Systems Sciences (HICSS03)*, pages 222–222, Hawaii, USA, 2002.
- [247] J. Wen, J. Nie, and H. Zhang. Clustering user queries of a search engine. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 162–168, New York, NY, USA, 2001. ACM Press.
- [248] P. Willet. Recent trends in hierarchical document clustering: a clisitcal review. *Information Processing and Management*, 24:577–597, 1988.
- [249] G. J. Williams and Z. Huang. Modelling the kdd process: A four stage process and four element model. Technical Report TR-DM-96013, CSIRO Division of Information Technology, February 1996.
- [250] C. Wong, S. Shiu, and S. Pal. Mining fuzzy association rules for web access case adaptation. In *In Workshop on Soft Computing in Case-Based Reasoning Research and Development, Fourth Int. Conf. on Case-Based Reasoning (ICCBR 01)*, 2001.
- [251] Y.H. Wu and A. L. Chen. Prediction of web page accesses by proxy server log. *World Wide Web*, 5(1):67–88, 2002.
- [252] J. Xiao, Y. Zhang, X. Jia, and T. Li. Measuring similarity of interests for clustering web-users. In *ADC '01: Proceedings of the 12th Australasian conference on Database technologies*, pages 107–114, Washington, DC, USA, 2001. IEEE Computer Society.
- [253] Q. Yang, H.H. Zhang, and I.T.Y. Li. Mining web logs for prediction models in www caching and prefetching. *Knowledge Discovery and Data Mining*, pages 473–478, 2001.
- [254] F. Yuan, H. Wu, and G. Yu. Web users classification using fuzzy neural network. *Lecture Notes in Computer Science*, 3213(1):1030–1036, 2004.

- [255] K. Zechner. Fast generation of abstracts from general domain text corpora by extracting relevant sentences. In *Procs. Int. Conf. on Computational Linguistics*, pages 986–989, 1996.
- [256] T. Zhang and V. S. Iyengar. Recommender systems using linear classifiers. *J. Mach. Learn. Res.*, 2:313–334, 2002.
- [257] T. Zheng and R. Goebel. An ncm-based framework for navigation recommender systems. In *Procs. 19th Int. Conf. on Artificial Intelligence, workshop Multi-Agent Information Retrieval and Recommender Systems*, pages 80–86, July 2005.
- [258] N. Zhong. *Advances in Web Intelligence*, volume 2663, chapter Toward Web Intelligence, pages 1–14. Springer-Verlag GmbH, 2003.
- [259] J. Zhu, J. Hong, and J.G. Hughes. Using markov chains for link prediction inadaptive web sites. In *Proc. Int. Conf. of ACM SIGWEB Hypertext*, pages 298–300, 2002.

Index

- Access
 - path, 114, 119
 - pattern, 117, 148
- Adaptive
 - hypermedia, 184
 - interface, 193
 - web site, 6, 125, 193, 195
 - web-based system, 120, 147
- Aggregation point, 69
- Algorithm
 - back propagation, 40
 - Max Flow-Min Cut, 101
 - PageGather, 120, 154
 - PageRank, 98
 - training SOFM, 171
- Arpanet, 2
- Artificial Neural Networks (ANNs), 37
- Association rules, 33, 113
- Bayesian network, 46
- Bitmap index, 74
- Business application server (BAS), 79
- Business to Business (B2B), Consumer (B2C), 5
- Caching, 116
- Classification, 34, 103, 112
- Client/server paradigm, 10, 60
- Cluster centroid, 36, 114, 175, 211
- Clustering, 34, 105, 110, 170, 209
 - density based, 36
 - hierarchical, 36
 - partition, 35
- Collaborative filter, 128
- Constellation model, 73
- Cookie, 12, 191, 198
- Cube
 - model, 70
 - pivoting, slicing, dicing, drill-down, roll-up, 70
- Customer Relationship Management (CRM), 78
- DARPA, 2
- Data
 - cleaning, 30, 56, 79
 - consistency, 29
 - consolidation, 27, 30, 79
 - irrelevant, 29
 - manipulation errors, 29
 - mining, 6, 28, 32
 - artificial neural networks (ANN), 37

- association rules, 33, 113
- bayesian network, 46
- classification, 34, 103, 112
- clustering, 34, 105, 110
- decisions trees, 44
- k-means, 43, 212
- k-nearest neighbor (KNN), 48
- self organizing feature maps (SOFM), 41
- self organizing feature maps (SOFM), 170, 209
- support vector machines (SVM), 50
- privacy, 144
- staging area, 31, 89, 203
- storage, 30, 86
- summarization, 27
- web, 179, 197
- web house, 31, 77
- Data base manager system (DBMS), 66
 - Multidimensional (MDBMS), 70
 - Relational, 70
- Data Warehouse, Mart, 28, 65
- Decision support system (DSS), 60, 188
- Decisions trees, 44
- Density Function, 35
- Dimensional table, 72, 191, 201
- Distance
 - cosine, 105, 165
 - Levenshtein, Edit, 168
- E-business, 1, 173
- ETL, 31, 66, 75, 186
- Granularity, grain, 70
- Grid, 61
- Hierarchies, 69
- Hyper Text
 - Transfer Protocol (HTTP), 10
 - Markup Language (HTML), 11
- Information
 - repository for web data, 186, 188, 196
 - systems, 63, 116
- Internet, 3
 - service provider (ISP), 11, 109
- IP Address, 109
- IP address, 13, 81, 159, 201
- K-means, 43, 212
- K-nearest neighbor (KNN), 48
- Kernel Function, 35
- Knowledge
 - base, 186, 189, 229
 - discovery from databases, 7, 25
 - repository, 183
 - representation, 180
- Learning
 - rate, 38
 - supervised, 34
 - unsupervised, 34, 171
- Lost in hyperspace, 125
- Materialized view, 74

- Max Flow-Min Cut, 101
- MDBMS, 70
- MDM
 - Snowflake, 73
- Measure
 - additive, 68, 188, 200
 - non-additive, 68, 190
 - semi-additive, 68
- Metadata, 29
- Mining
 - data, 6, 28
 - web, 93, 94
 - web content, 102
 - web structure, 94
 - web usage, 109
- Multidimensional
 - data model (MDM), 68
 - granularity, grain, 70
 - analysis, 67
 - array, 71
 - Data base manager system (MDBMS), 70
- Multilayer, 39
- Neighborhood kernel, 42
- Neuron neighborhood, 36, 171
- One-to-One marketing, 133
- Online Analytical Processing (OLAP), 25, 60, 67, 188
 - relational (ROLAP), 86
- Operational system, 25, 60
- Page Interest Estimators (PIE), 112
- PageGather, 120
- Partition table, 74
- Peer to Peer (P2P), 5
- Perceptron, 38
- Personalization, 133
 - Consumer-centric, 137
 - Provide-centric, 137
- Portal, 5, 125
- Predictive model, 28
- Prefetching, 116
- Privacy, 144
- RAID, 61
- RDBMS, 70
- Recommendation, 151
 - Computerized system, 185
 - offline, 118, 217
 - online, 118, 220
 - system, 126
 - testing, 220
 - web-based system, 132
- Relationship Management (URM), 78
- Repository
 - information, 59, 186
 - knowledge, 189, 196
 - ratterns, rules, 188
- Response time, 66
- Self Organizing Feature Maps (SOFM),

- 41, 170, 209
- Sessionization, 15, 162, 201
 - heuristics
 - navigation,time oriented, 16
 - strategies
 - proactive, reactive, 16
- Sigmoid, 39
- Similarity measure, 35, 111, 162, 207
- Snowflake, 73
- Social network, 6, 101
- SOFM, 170, 209
- Star model, 72, 190, 196, 201
- Star query, 73, 233
- Stemming process, 103, 106, 164, 173
- Structural Risk Minimization (SRM), 52
- Support vector machines (SVM), 50
- Uniform Resource Locator (URL), 10
- User
 - amateur,experienced, 170, 221
 - behavior, 6, 208
 - business, 60
 - preferences, 6, 174, 213
 - Privacy, 129, 136, 141
 - privacy, 144
 - Relationship Management (URM), 78
 - session, 15, 18
 - real, 18
- Vector
 - important page, 174, 214
- space model, 19, 164, 206
- user behavior, 158
- Web, 1
 - browser, 10
 - community, 21, 94, 101
 - content mining, 102, 213
 - crawler, 15
 - customer, 2
 - data, 9, 26, 59, 179, 197, 200
 - hyperlink structure, 21, 87, 94
 - Information repository, 186, 196
 - information repository, 59, 188
 - intelligence, 6
 - log files, 13, 88, 198
 - master, 29, 59, 120
 - mining, 55, 93
 - objects, 12
 - operation, 10
 - page, 10, 19
 - authoritative, 95
 - authorities, 22
 - hub, 22, 95
 - text summarization, 107
 - personalization, 139
 - query mining, 115
 - server, 10
 - site, 1, 6, 10, 125, 198
 - site keywords, 108, 173, 213
 - structure mining, 94
 - clever, 98

- hits, 95
- Page Rank, 98
- usage mining, 109
- user, 2, 133, 185
 - behavior, 56, 141, 158, 212
- visitor, 2, 81, 109, 130, 185, 198
- W3, 3
- World Wide, 1